Methods and tools for the statistical data analysis of large datasets collected from bio-based manufacturing processes

Spooner, Max Peter

Publication date:
2018

Document Version
Publisher's PDF, also known as Version of record

PhD Thesis
Doctor of Philosophy

DTU Compute
Department of Applied Mathematics and Computer Science

Methods and tools for the statistical data analysis of large datasets collected from bio-based manufacturing processes

Max Spooner

Kongens Lyngby 2018

DTU Compute
Department of Applied Mathematics and Computer Science
Technical University of Denmark

Matematiktorvet
Building 303B
2800 Kongens Lyngby, Denmark
Phone +45 4525 3031
compute@compute.dtu.dk
www.compute.dtu.dk

Summary (English)

In bio-manufacturing, biological systems are harnessed for the production of useful organic materials to be used in, for example, the food, medicine or agricultural industries. The most common mode of production in this sector is through batch processes. In a batch process a reactor vessel is filled with raw materials such as bacteria culture, water and sugar. It is then subjected to controlled conditions for a finite duration during which its contents undergo transformation, and finally the end-product is harvested from the reactor. Typically, a variety of sensors measure conditions in the reactor throughout each batch, such as temperature, pressure and concentration. With advances in sensor technology, and computational power, the volume of data collected in this way is ever increasing. The goal of the thesis is to contribute new techniques for utilising this data to improve process understanding and product quality.

The existing literature on statistical monitoring, and quality prediction, for batch processes is reviewed, highlighting the challenges presented by batch process data. These include its three dimensional structure (conventionally represented as I batches $\times$ J variables $\times$ K time-points) comprising highly multivariate, cross-correlated, auto-correlated and non-stationary variable trajectories for each batch. An aspect of the data which leads to a number of contributions in the thesis is the variation in the time dimension often present in batch processes, meaning that comparable events occur at different times in different batches, so that the shapes and features in the resulting variable trajectories are not synchronised. In addition, the overall duration of different batches in a process may vary leading to different numbers of observations, complicating the application of standard bi-linear or tri-linear methods. Dynamic time warping (DTW) has previously been applied to synchronise batch process data and address these issues. The DTW algorithm identifies an optimal warping function, which stretches and compresses each batch in order to synchronise the variable trajectories. The warping function obtained for each batch may be interpreted as the

progress signature of the batch. Using a case study of a bacteria culture batch process from Chr. Hansen, the advantages of including local constraints in the DTW algorithm, so that the warping function is a more realistic representation of batch progress, are demonstrated, and a method for selecting the local constraint is presented.

In another case study using data from Chr. Hansen, a novel method is developed for predicting the harvest time of a batch at an early stage, whilst the batch is in progress. The method utilises lasso regression for selection of important variables for making the prediction, and combines the prediction with the progress information contained in the warping function from online alignment with DTW. Early harvest time prediction can contribute to scheduling of down-stream resources. In a third real industrial case study, lasso regression is again utilised to obtain quality predictions for batches of pectin produced by CP Kelco. The approach is contrasted with partial least squares models, and comparable estimated prediction error is obtained using lasso regression, in addition to a more parsimonious and interpretable model.

Finally, the ability of DTW to quantify similarity between time series is exploited to develop a method for monitoring batch processes online to detect if a fault occurs. This method is based on the nearest neighbour principle, comparing an ongoing batch to its k nearest neighbours in a database of successful batches, according to the DTW distance. If the distance to the k nearest neighbours increases too quickly, an alarm is signalled to indicate that a fault has occurred. The method is demonstrated using a simulated dataset, representing batch production of penicillin, which contains a wide variety of fault types, magnitudes and onset times. The performance of the novel method is contrasted with a benchmark principle component analysis based approach, and shown to have a higher detection rate and faster detection speed when there is clustering of batches in the reference dataset.

Summary (Danish)

I biologisk produktion, udnyttes levende systemer til produktion af nyttige organiske stoffer, for eksempel i fødevare- og medicinal-industrien eller landbruget. Den hyppigst anvendte produktionsmetode i denne sektor er batchprocesser, som begynder med tilførsel af en mængde råstoffer til en reaktor. Under kontrollerede forhold, og gennem et begrænset tidsinterval, gennemgår råstofferne en transformation, og det færdige produkt tages til sidst ud af reaktoren. Ofte måles forholdene i reaktoren, såsom temperatur, tryk og koncentration, undervejs af sensorer. Med den konstante udvikling af sensorteknologi og beregningskraft, øges mængden af indsamlet data hele tiden. Målet med denne afhandling er at præsentere nye metoder hvorved disse data kan udnyttes til at fremme forståelsen for processen og produktkvalitet.

Den eksisterende litteratur om statistik overvågning og kvalitetsforudsigelse, af batchprocesser bliver gennemgået. Udfordringer med batchprocesdata inkluderer den tredimensionelle struktur som indeholder multivariate, korrelerede, auto-korrelerede, ikke-stationære variabler. Variation i tidsdimensionen forårsager ofte at de samme begivenheder finder sted på forskellige tidspunkter i forskellige batches. Den overordnet varighed af batches kan ligeledes variere, hvilket vanskeliggør brug af de sædvanlige bi- eller tri-lineære metoder. Dynamic Time Warping (DTW) er tidligere blevet brugt til synkronisering af batchprocesdata. DTW algoritmen identificerer den bedste måde at forvrænge tidsaksen i hver batch, hvorved den tætteste synkronisering af variabelprofilerne er opnået. Måden hvorpå hver batch forvrænges på kan betragtes som en slags progressionssignatur for hver batch. Via et casestudie med data fra Chr. Hansen, som omhandler en batchproces til produktion af bakteriekultur, undersøges brug af lokale begrænsninger i DTW algoritmen, som bidrager til en mere realistisk progressionssignatur. En metode til selektion af lokal begrænsning præsenteres.

I et andet casestudie med data fra Chr. Hansen, præsenteres en ny metode til forudsigelse af det endelige udtagelsestidspunkt på et tidligt stadie i processen. Denne

metode indebærer lasso regression til udvælgelse af væsentlige variabler. Forudsigelsen fra lasso modelen bliver kombineret med tidsinformation fra DTW algoritmen. Tidlige prognoser af batchvarighed er nyttige i forhold til skemaplanlægning samt allokering af ressourcer. I et tredje casestudie fra industrien, anvendes lasso regression til forudsigelse af produktkvalitet i en pektinfremstillingsproces med firmaet CP Kelco. Metoden er sammenlignet med en Partial Least Squares model. En lignende præcision er opnået med lasso metoden, som desuden er enklere og nemmere at forstå.

Endelig bliver DTW algoritmens evne til at kvantificere ligheden mellem tidsrækker udnyttet i en ny overvågningsmetode til batchprocesser. Denne metode er baseret på "k nearest neighbours" princippet. En igangværende batch sammenlignes med dennes k nærmeste naboer i en database af vellykkede batches, ud fra DTW distancen. Hvis distancen øges for hurtigt, genereres en alarm som indikerer at en fejl har fundet sted. Metoden demonstreres på et simuleret datasæt, som repræsenterer batchprocessen ved produktion af penicillin. Datasættet indeholder batches med en bred vifte af fejltyper i forskellige størrelsesordner. Metoden sammenlignes med den klassiske PCA baserede metode. Den nye metode er mere følsom overfor fejl i processen, og signalerer hurtigere, når der findes grupperinger mellem batches.

Preface

This PhD thesis was prepared at the department of Applied Mathematics and Computer Science at the Technical University of Denmark in fulfilment of the requirements for acquiring a PhD in Applied Mathematics. The PhD was partly financed by the BioPro2 project.

The thesis deals with statistical methods for analysis of production data from biomanufacturing. Through case studies using real production data from Chr. Hansen and CP Kelco, as well as a simulated case study of penicillin production, methods are presented for aligning data, predicting batch harvest time, predicting batch quality and monitoring batch processes.

The thesis is structured around three research papers, with additional chapters providing greater context. All of the work comprising this thesis was completed during the study period.

Kongens Lyngby, November 20, 2018

Max Spooner

List of publications

The following papers are included in the thesis:

- Paper 1: Spooner, M., D. Kold, and M. Kulahci. 2017. "Selecting local constraint for alignment of batch process data with dynamic time warping". *Chemometrics and Intelligent Laboratory Systems* 167:161–170. https://doi.org/10.1016/j.chemolab.2017.05.019.
- Paper 2: Spooner, M., D. Kold, and M. Kulahci. 2018. "Harvest time prediction for batch processes". *Computers & Chemical Engineering* 117:32-41. https://doi.org/10.1016/j.compchemeng.2018.05.019.
- Paper 3: Spooner, M., and M. Kulahci. 2018. "Monitoring batch processes with dynamic time warping and k-nearest neighbours". *Chemometrics and Intelligent Laboratory Systems* 183:102-112. doi: https://doi.org/10.1016/j.chemolab.2018.10.011

Work related to these papers was presented at the following conferences:

- Spooner, M., and M. Kulahci. "Selecting appropriate constraints for alignment of batch process data with dynamic time warping", *The Fourth International Conference on the Interface between Statistics and Engineering*, Palermo, Italy, June 2016. (Abstract and presentation)
- Spooner, M., and M. Kulahci. "Harvest time prediction for batch processes", *The 17^{th} Annual Conference of the European Network for Business and Industrial Statistics*, Naples, Italy, September 2017. (Abstract and presentation)
- Spooner, M., and M. Kulahci. "Fault detection for batch processes using k nearest neighbours and dynamic time warping", *The 18^{th} Annual Conference of the European Network for Business and Industrial Statistics*, Nancy, France, September 2018. (Abstract and presentation)

Acknowledgements

I would like to thank my supervisor, Murat Kulahci, for all of his advice and words of encouragement throughout the PhD. Your enthusiasm and positive outlook was a constant motivation. Thank you to my co-supervisor, Bjarne Ersbøll, who was always ready to step in when needed. Thank you to Line Clemmensen who shared her knowledge and experience during her time as my co-supervisor.

It has been a privilege working together with the companies Chr. Hansen and CP Kelco. Thank you to Lisbeth Grubov, David Kold and colleagues at Chr. Hansen for their commitment to the project and many interesting discussions. Also thank you to Paloma Santacoloma, Tommy Jensen and colleagues at CP Kelco for a rewarding collaboration. My thanks to all collaborators in the BioPro2 framework who I have spoken to at one time or another and exchanged ideas with, including managing director Jesper Bryde-Jacobsen who was always open to new ideas.

Thank you to the research team in the Chemical Engineering Department at UC Davis who were welcoming and supportive during my stay there in the Spring of 2016.

Finally, thank you to my colleagues in the section for Statistics and Data Analysis for a friendly and rewarding working environment and thanks to my friends and family for their love and support.

Contents

Summary (English) i

Summary (Danish) iii

Preface v

List of publications vii

Acknowledgements ix

Contents xi

1 Introduction 1
- 1.1 Objectives . . . 2
- 1.2 Contributions . . . 3
- 1.3 Outline of the thesis . . . 4

2 Review of the literature 5
- 2.1 Batch processes . . . 5
- 2.2 Monitoring . . . 8
  - 2.2.1 MPCA . . . 8
  - 2.2.2 Comments on MPCA . . . 11
  - 2.2.3 Variable-wise unfolding and related methods . . . 12
  - 2.2.4 Three-way methods . . . 14
  - 2.2.5 Alignment of batch process data . . . 15
  - 2.2.6 Closing remarks on monitoring . . . 22
- 2.3 Prediction . . . 23
  - 2.3.1 Partial Least Squares based methods . . . 24
  - 2.3.2 Other regression based methods . . . 25
  - 2.3.3 Non-linear methods . . . 27
- 2.4 Chapter conclusion . . . 27

3 Aligning batch data with Chr. Hansen 29
- 3.1 Paper 1: Selecting local constraint for alignment of batch process data with dynamic time warping . . . 32

4 Predicting batch harvest time with Chr. Hansen **43**
4.1 Paper 2: Harvest time prediction for batch processes 44

5 Predicting pectin quality with CP Kelco **75**
5.1 Introduction 75
5.2 Data preprocessing and exploratory data analysis 78
5.2.1 Process data 78
5.2.2 Batch background data 84
5.2.3 Response data 85
5.3 Results and discussion 85
5.3.1 Model fitting 86
5.3.2 Estimation of prediction error 89
5.4 Chapter conclusion 90

6 A new method for monitoring batch processes **93**
6.1 Paper 3: Monitoring batch processes with dynamic time warping and k-nearest neighbours 94

7 Conclusion **129**

Bibliography **133**

CHAPTER 1

Introduction

Bio-based manufacturing is a flourishing industry in Region Zealand, Denmark. Here, a variety of biological systems have been harnessed for commercial production of a range of products. For example, Chr. Hansen produces bacteria cultures, enzymes, and other bio-products for use in food, medicine and agricultural industries. CP Kelco produces food and medicine products based on plant-derived pectin and carrageenan. Novo Nordisk produces insulin for treatment of diabetes, and Novozymes produces enzymes and probiotics for agriculture and biofuel industries. Finally, Ørsted produces bio-ethanol from fermentation of hay and straw. BioPro2, a comprehensive project consisting of a collaboration among the companies just named, researchers at the Technical University of Denmark and the University of Copenhagen, and management from Region Zealand, was set up to strengthen this biotech cluster and foster innovation and efficiency. The PhD project documented in this thesis is one of the many elements within the BioPro2 framework.

One area in which to pursue innovation and efficiency in bio-manufacturing, is in the huge amounts of production data it now generates. Improvements in sensor technology, and computational technology in general, have made the collection, storage and processing of ever increasing amounts of production data both technically possible and affordable, a phenomenon which has been described by the ubiquitous term ”big data”. However, although sectors such as internet retail, social media and entertainment services have utilised big data analytics for some time, the manufacturing industry has been less quick to explore its potential. Often, even though process data is intensively collected and stored by manufacturing companies, and may be used for spot-checking product quality and production status, it is not used for discovering wider patterns that may improve quality and add value to the company.

The term process data is here used to refer to the variables measured over time during the batches of a particular bio-manufacturing process, and typically includes, for

example, temperature, pH, various substance concentrations and flow rates. These engineering variables may be distinguished from the more direct chemical analysis data commonly dealt with in the field of chemometrics, which can include, for example, infra-red and mass spectroscopy data. Chemometrics has made a valuable contribution to bio- as well as chemical manufacturing, and spectroscopic analysis is now wide-spread in the industry, enabling companies to gain insight into the chemical composition of materials using analysis of wavelength data. However, the focus of this thesis is on the process engineering variables. In most processes, this data is freely available as the engineering variables are measured for controlling the conditions in the process, and the aim is to extract more value from this existing resource rather than implement new measurement systems.

The overall aim of the PhD project was therefore to investigate methods for extracting useful information from the abundant data produced by bio-manufacturing, and thereby contribute practical value to both the academic literature on this subject, and the industry partners.

1.1 Objectives

The primary objective of the thesis is to contribute new methodologies for gaining insight from bio-manufacturing process data, that address the specific challenges of this data. The goal of these methodologies is to improve product quality, and contribute to greater process understanding. The challenges of bio-manufacturing data result from its mode of production in batches. As batches of raw materials are transformed into the finished products, process data takes on a three dimensional structure with batches, variables, and time-points, forming the three dimensions. Processes are typically not at steady-state, leading to dynamics in the variable trajectories. Often many variables are measured, so there is high dimensionality. Cross-correlation between variables, and auto-correlation over time are usually present. In bio-manufacturing, the time dimension of the data can be a big source of variation between batches. This is due to the biological basis of the processes, which introduces variation in, for example, growth rates. The living element in bio-manufacturing can be an extra source of uncertainty compared to more deterministic processes in chemical and physical manufacturing. Therefore, in developing methodologies, contributions should be made to addressing some of these challenges.

A second goal of the thesis is to apply methods to real case studies in bio-manufacturing. There are many existing methodologies for the exploration of batch process data, but practical lessons from applying the methods to real cases are needed. By investigating real industrial cases, theoretical tools that solve the problems actually encountered in industry may be developed.

Finally, it is a goal of the thesis to document the exchange of knowledge between

industry and academia. There is a wealth of literature on theoretical approaches to analysis of batch process data, but naturally there is a time-lag in implementing state of the art research in industry. Through close collaboration, it is the aim to disseminate state of the art research to industry partners. In return, it is the aim to learn about the practical problems encountered in bio-processes and raise awareness of these problems, and the solutions found, in the academic community. In this way, value may be contributed to both academia and industry.

1.2 Contributions

This thesis collects a number of significant contributions from the PhD project. The first contributions resulted from a fruitful collaboration with Chr. Hansen. Here, substantial time variation was observed in the batch fermentations of bacteria cultures, and this motivated the focus of research towards alignment of batch process data. The purpose of alignment is to ensure comparable events are synchronised, and to obtain the same amount of data for each batch. Dynamic time warping (DTW) is an alignment method which has been adopted by many researchers for alignment of batch process data. This method can extract the time variation from batches by identifying a warping function representing the progress signature of a batch. A need was identified for improving the existing approach to DTW alignment of batch processes so that more realistic alignments were obtained. Using data from Chr. Hansen, the use of so-called local constraints in DTW were investigated, and a novel method was developed for selecting the local constraint to use. This contribution forms the basis of "Paper 1" (Spooner, Kold, and Kulahci 2017).

Further collaboration with Chr. Hansen lead to another contribution based on DTW. Variation was observed in the time at which different batches were stopped, the so-called harvest time, motivating an investigation into ways of predicting the harvest time at an early stage in a batch. If the harvest time could be predicted in advance, then batches could be harvested more consistently which would hopefully lead to improvement in quality. In addition, early prediction of harvest time could facilitate scheduling of resources. A novel method was developed for using the warping function from DTW to predict harvest time of a batch. This method was combined with lasso regression on the process data, in order to perform automatic variable selection so that the variables of greatest relevance for harvest time were selected. These contributions are presented in "Paper 2" (Spooner, Kold, and Kulahci 2018).

Another interesting collaboration was undertaken with CP Kelco that produces various pectin products for the food industry. Different varieties of pectin are characterised by two parameters: Degree of amidation and degree of esterification. Through a batch process known as amidation, these parameters can be changed to produce a specific product. A project was started with the aim of predicting degree of amidation and degree of esterification using process data. Again, lasso regression was applied,

due to its variable selection capability which leads to easier model interpretation. In the literature, latent structure methods like partial least squares are more commonly applied for modelling batch data. The promotion of lasso regression using real case studies, as an alternative to latent structure methods, is a contribution resulting from the project with CP Kelco. Some challenges were encountered related to using production data for discovery of new causal relationships, and highlighting lessons related to these challenges is another contribution of the thesis.

Finally, in order to focus on developing a new methodology for monitoring and detecting faults in batch bio-processes, research using a simulated dataset was carried out. This ensured a more controlled setting so that theoretical aspects of the method could be investigated. A data driven approach based on DTW and k Nearest Neighbours was developed, where an ongoing batch is compared to a historical dataset of successful batches. The similarity between the ongoing batch and the reference dataset is quantified by the DTW distance to the k nearest successful batches in the dataset. If the distance increases too quickly an alarm is signalled indicating that a fault has occurred. This novel method for monitoring batch processes is presented in "Paper 3" (Spooner and Kulahci 2018).

1.3 Outline of the thesis

The thesis is structured to support the three papers which constitute the main contributions. Chapter 2 provides a review of the literature related to statistical data analysis in bio-manufacturing and batch processes in general, with special focus on those methods which lead to new contributions in the papers. Chapters 3 and 4 contain the results from the collaboration with Chr. Hansen, and include Paper 1 and Paper 2 respectively. Before the papers, an introduction to each chapter provides context to the contribution. In Chapter 5, results and lessons learnt from the collaboration with CP Kelco are presented. Chapter 6 contains Paper 3, again with introductory text providing the context for this paper. Finally, the key findings from the thesis are summarised in the concluding chapter, together with ideas for future work, and lessons learnt from the PhD.

CHAPTER 2
Review of the literature

In this chapter selected literature related to the statistical analysis of bio-based manufacturing processes will be reviewed. An overview of the available methods in this field is presented. Those methods which formed the basis for the novel contributions of the thesis are presented in depth.

2.1 Batch processes

In bio-based manufacturing batch processes dominate as the chosen mode of production (Croughan, Konstantinov, and Cooney 2015), and all the cases encountered during this project are batch processes. Batch production is one of the oldest forms of production, with early examples being the baking of bread and fermenting of wine in batches. Batch processes are characterised by the following three steps (Felder and Rousseau 2003; Ündey and Çinar 2002)

1. Start: A vessel is charged with raw materials
2. Transformation: The vessel contents are subjected to controlled conditions for a finite duration.
3. End: The product is harvested (removed) from the vessel

In the baking example, dough (the raw material) is placed in the oven (the vessel), and subjected to controlled temperature conditions, after which the product (loaf of bread) is removed from the oven. The concept of batch processing may be better understood by comparing it to its main alternative: continuous processing. Continuous processes are a mode of production which run without interruption for long periods at a time. Raw materials are constantly fed into the system, conditions are kept at a steady state

such that the raw materials are continually transformed, and the resulting product is continually removed from the system (besides infrequent maintenance shut-downs). Refining of crude oil is an example of a typical continuous process, where crude oil is continuously fed into a distillation column, and the various end products (petrol, kerosene, diesel etc.) are continuously extracted, in a steady state which can run for long periods without interruption (Alfke, Irion, and Neuwirthe 2007). Continuous processes are suitable for production of an identical product over long periods of time where demand for the product is reasonably constant. In contrast, batch processes are ideal for producing varieties of speciality products. The same equipment can be used for producing different versions of a product depending on the wishes of different customers. However, batch processes are less efficient than continuous processes as in-between batches, equipment remains idle.

Croughan, Konstantinov, and Cooney (2015) offer some suggestions why batch processes still dominate in industrial bioprocessing, despite the greater efficiency of continuous processes. These reasons are closely linked to the *living* element present in bioprocesses and include: the difficulty of ensuring homogeneous conditions in the reactor vessel in order to avoid regions of cell accumulation and regions of cell death; the need for periodic sterilisation of equipment; and the inherent complexity of biological systems. Batch processes are also more suited to smaller scale production of speciality products, whilst continuous processes are ideal for large scale production of homogeneous product. Hence, the subject of data analysis for bioprocesses is closely related to data analysis of batch processes in general.

Statistical data analysis has generally been applied to batch processes to solve two broad problems. The first is the problem of process monitoring. Here, the aim, as in the well established field of statistical process control, is to determine whether a process is in control, or out of control, based on data measured during the process. The second is the problem of quality prediction: predicting some quality parameter, e.g. product yield, based on the available input data from the batch process. Of course these are not problems unique to batch processes. However, certain properties of batch process data mean that solutions specific to this domain were needed. These two problems form the topics of the following two sections.

For greater insight into the methods reviewed in this chapter, a dataset from a simulated batch process will be used for illustrative purposes. This dataset, generated by Van Impe and Gins (2015) based on the PenSim model framework of Birol, Ündey, and Çinar (2002), represents batch production of penicillin. Data representing 50 batches of this process is shown in Fig. 2.1.

The data from batch processes has a three way structure, where, for I batches, J variables are measured at K_i time-points (where batches vary in duration, K_i may vary from batch to batch). For example, the penicillin dataset in Fig. 2.1 shows $J = 12$ variables measured for $I = 50$ batches at $455 \leq K_i \leq 467$ time-points. There

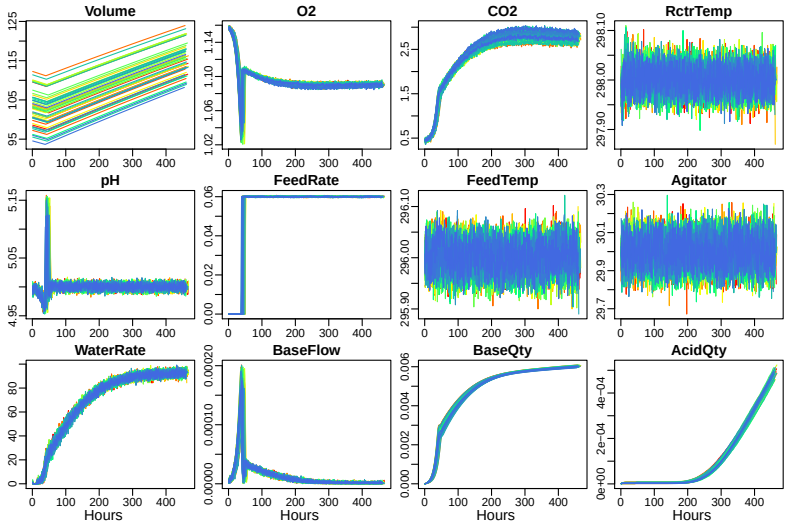

Figure 2.1: Process data for 50 batches of penicillin (see Table 1 in Chapter 6 for the full variable names and units of measurement).

are challenges of monitoring and prediction for batch processes associated with each mode of this three-way data structure. In the variable dimension it is common for a large number of variables to be measured, and these variables are usually correlated with each other (Nomikos and MacGregor 1994) (note, for example the similarity of CO2, WaterRate and BaseQty in Fig. 2.1). The 12 variables in the penicillin example is a mild case - in other batch processes there can be 30 or more variables with some quantities, e.g. temperature, measured at several locations in the reactor and therefore highly correlated. With respect to the time dimension, the main challenge is that batch processes are by definition unsteady-state meaning that the process variables change with time during a batch, resulting in variable trajectories. The variables are therefore autocorrelated, and correlation between variables may change throughout a batch (Nomikos and MacGregor 1994). A pH of 5.15 in the penicillin example has extremely different implications depending on the current time in the process (Fig. 2.1). Techniques developed for steady-state (i.e. continuous) processes are therefore not well suited to batch processes. Also related to the time dimension is the challenge that both the overall durations, and the timing of different events within batches, often varies (González-Martínez et al. 2014). Finally, in the batch dimension a challenge is that variation of the different variables between batches may be skewed (due, for example, to the nature of measured variables like concentrations) or clustered (due, for example, to changes in suppliers of raw materials) (Van Impe and Gins 2015).

The preceding summary of some of the challenges posed by batch process data provides a context for the methods of monitoring and prediction reviewed in the following sections.

2.2 Monitoring

Nomikos and MacGregor (1994) formulated the problem of statistical process monitoring of batch processes and presented a solution which has been actively adopted and/or developed by many authors since its introduction. Previous work regarding monitoring of batch processes were mainly based on mechanistic models of the process (e.g. differential equations describing rates of change of quantities in the process), or rule-based expert systems (devising algorithms that make use of the knowledge of process engineers and operators). Nomikos and MacGregor instead took a multivariate statistical approach using only the process data itself, together with the statistical control chart philosophy founded by Shewhart (1931) to formulate a method for monitoring batch processes using multiway principal component analysis (MPCA). In summary, their approach entails unfolding the three-way batch process data into the more familiar two-way matrix structure such that the standard (Montgomery 2013) multivariate statistical process monitoring tools of PCA with associated Hotelling T^2 and Q control charts, can be used. As the work of Nomikos and MacGregor (1994) has been influential, an outline of the method is presented.

2.2.1 MPCA

Let $\underline{\mathbf{X}}$ (I batches $\times$ J variables $\times$ K timepoints) represent the data-cube formed by I completed batches of a process (Nomikos and MacGregor assume batches of equal duration, K) and suppose these batches are known to be in control. Then MPCA can be used to build a model to describe this data. The model can be applied to a new batch whilst it is in progress, to determine if it is progressing in accordance with normal operating conditions, or should be declared out of control. An out of control alarm indicates that some fault has occurred. To build the model, first the historical data $\underline{\mathbf{X}}$ is unfolded by placing each vertical slice side by side, resulting in the matrix $\mathbf{X}^*$ ($I \times JK$). This type of unfolding has become known as "batch-wise" unfolding, to distinguish it from "variable-wise" unfolding considered, for example, by Wold et al. (1998). Both unfolding methods are shown in Fig. 2.2. They are so-called because in batch-wise unfolding the batch dimension, I, is unchanged, and in variable-wise unfolding the variable dimension J is unchanged. Each column of $\mathbf{X}^*$ is centered and scaled by subtracting its mean and dividing by its standard deviation resulting in $\mathbf{X}$ (the column means and standard deviations are saved for future application of the model). Here, mean-centering is equivalent to subtracting the average trajectory from each variable, and Nomikos and MacGregor claim it removes most of the nonlinear behaviour from the data. Next, PCA is applied to decompose the scaled and centered $\mathbf{X}$ into scores, $\mathbf{T}$ ($I \times R$) and loadings, $\mathbf{P}$ ($JK \times R$), where R is the number

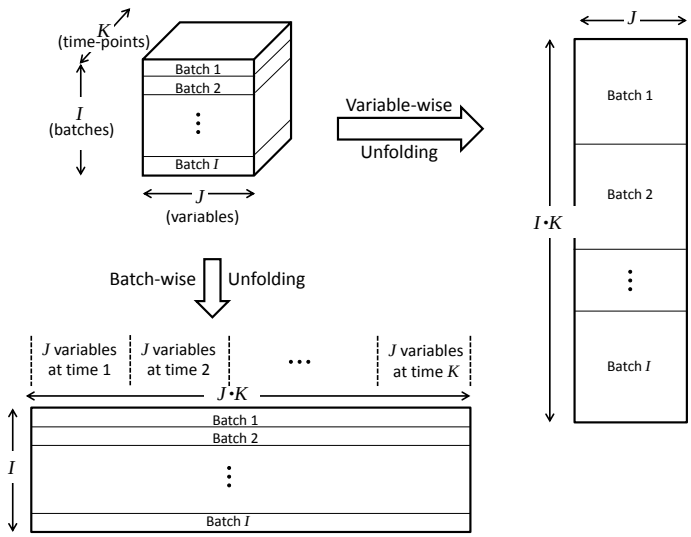

Figure 2.2: Batch-wise unfolding and variable-wise unfolding.

of components retained in the model. Thus

$$\mathbf{X} = \mathbf{T}\mathbf{P}' + \mathbf{E} \tag{2.1}$$

where $\mathbf{E}$ $(I \times JK)$ contains the residuals. The loading vector contained in the i^{th} column of $\mathbf{P}$ is the eigenvector corresponding to the i^{th} largest eigenvalue of the correlation matrix of $\mathbf{X}$. The R loading vectors define the R directions of greatest variation in the data, and the scores give the coordinates of each batch in this coordinate system. Thus, the PCA decomposition implies that the data can be explained by an R dimensional subspace, with any deviation from this subspace contained in $\mathbf{E}$ and attributed to noise. The number of components to retain in the model can be selected using cross validation, or from visual inspection of the scree plot of the eigenvalues (the broken stick rule).

Suppose a new batch has just been completed and one wishes to apply the model (the case of online monitoring during a batch will be discussed presently). Then the new batch is unfolded, centered and scaled to form the $JK \times 1$ vector $\mathbf{x}$, and the scores calculated as

$$\mathbf{t} = \mathbf{x}'\mathbf{P} \tag{2.2}$$

At time k of an ongoing batch each variable trajectory is only known up until time k. It is therefore necessary to fill in the future values in order to construct

$\mathbf{x}$ and apply Equation 2.2. Nomikos and MacGregor (1995b) propose three infilling methods. In all cases, those rows of $\mathbf{x}$ corresponding to times less than or equal to k are constructed as usual

1. Zero infilling: Rows corresponding to times greater than k are set to 0. This assumes that the future variable trajectories coincide exactly with the mean trajectories from the reference dataset used for centering. Then Equation 2.2 is applied to obtain the current scores.

2. Constant deviations: Assume each variable will deviate from the mean by the current deviation for the remainder of the batch. Then Equation 2.2 is applied to obtain the current scores

3. Projection to the model plane: Treat the future values like missing values, and use the ability of PCA to impute missing data, i.e., let $\tilde{\mathbf{P}}$ $(Jk \times R)$ and $\tilde{\mathbf{x}}$ $(Jk \times 1)$ be reduced versions of $\mathbf{P}$ and $\mathbf{x}$ respectively, such that they only contain those rows of $\mathbf{P}$ and $\mathbf{x}$ corresponding to times less than or equal to k. Then the current scores are calculated by

$$\mathbf{t} = (\tilde{\mathbf{P}}'\tilde{\mathbf{P}})^{-1}\tilde{\mathbf{P}}'\tilde{\mathbf{x}} \tag{2.3}$$

Nomikos and MacGregor (1994) outline construction of two control charts for monitoring two summary statistics from the model. The first is a Hotelling's T^2 chart:

$$T^2 = \mathbf{t}\mathbf{S}^{-1}\mathbf{t}' \tag{2.4}$$

where $\mathbf{S}$ $(R \times R)$ is the sample covariance matrix of the score vectors $\mathbf{t}$ from the reference data-base for time k with the chosen infilling method. It is necessary to use a time-dependent covariance matrix because the online scores are time dependent due to the necessity of infilling. Assuming multivariate normality of the scores, significance level α control limits for T^2 are given by

$$T^2_\alpha = \frac{R(I^2-1)}{I(I-R)} F_{(R,I-R)}(\alpha) \tag{2.5}$$

where $F_{(R,I-R)}(\alpha)$ is the (1-α) quantile of the F distribution with R and $I-R$ degrees of freedom. Louwerse and Smilde (2000) point out a problem with estimating $\mathbf{S}$ from the scores of the reference dataset, in that these scores come from batches used to fit the PCA model, whereas the scores of a new batch are independent of the model. For the control chart to be accurate on new batches they therefore recommend a leave-one-out procedure for estimating $\mathbf{S}$, whereby each batch in the reference dataset is left out, a new PCA model is fit to the remaining batches and the online scores of the left out batch are calculated. Then $\mathbf{S}$ is calculated from these leave-one-out scores and

is therefore a more accurate estimate of the covariance of the scores of a completely new batch. This approach is preferable and is also adopted by, for example, Rato et al. (2016). The second control chart monitors the sum of squared errors, Q. The error vector at time k is given by

$$\mathbf{e} = \mathbf{x} - \mathbf{tP}' \tag{2.6}$$

Q is constructed as the sum of squared errors of only those errors corresponding to the current time k. That is, let $\tilde{\mathbf{e}}$ $(J \times 1)$ be the rows of $\mathbf{e}$ corresponding to time k, then

$$Q = \tilde{\mathbf{e}}'\tilde{\mathbf{e}} \tag{2.7}$$

The reason for only including the instantaneous errors in Q is so that they are not influenced by the "errors" of the filled-in portion of $\mathbf{x}$. Nomikos and MacGregor (1995b) show that it is reasonable to assume Q has approximately a weighted chi-squared distribution $g\chi^2_h$ where the weight g and degrees of freedom h can be estimated for each time point by the sample mean (m) and variance (v) of the Q values of the reference dataset as

$$g = \frac{v}{2m} \textbf{ and } h = \frac{2m^2}{v} \tag{2.8}$$

Then control limits at significance level α for the Q chart at time k are given by

$$Q_\alpha = g\chi^2_h(\alpha) \tag{2.9}$$

where $\chi^2_h(\alpha)$ is the $(1-\alpha)$ quantile of the χ^2 distribution with h degrees of freedom. Again, more accurate control limits are obtained by using the leave-one-out procedure to estimate g and h from the reference dataset as suggested by Louwerse and Smilde (2000).

Hotelling's T^2 quantifies the distance of a batch from the origin in the R dimensional subspace of the MPCA model, whilst Q represents the perpendicular distance of the batch to the model plane. If a new batch displays extreme variation, but in a manner similar to that seen in the reference dataset then T^2 should be large, whilst if the variation is of a type not seen in the reference dataset Q should be large. If the variation is sufficiently great, then Q or T^2, or both, will exceed their control limits and an alarm is signalled indicating that a fault has occurred.

Nomikos and MacGregor (1994) demonstrate the MPCA method using a simulated dataset representing batch polymerization of styrene-butadiene in which $J = 9$ variables are measured at $K = 200$ time-points for each batch. They use a reference dataset of $I = 50$ normal batches to build the MPCA model and demonstrate online monitoring of two faulty batches, and one normal batch.

2.2.2 Comments on MPCA

Nomikos and MacGregor (1994) introduced the philosophy of statistical process control to the batch process industry, and provided a rigorous statistical framework for

monitoring batch processes which influenced much of the literature that followed. The time dynamics of the variables are handled by unfolding, and thus combining the time dimension with the variable dimension. This leads to a very wide matrix with many highly correlated columns. The dimensions are reduced by PCA to R, the number of retained components.

A disadvantage of the method is that it assumes the reference dataset adequately represents normal operating conditions. If normal operating conditions change, due to, for example, a change in seasons, supplier or sensor drift, then the model will not be applicable unless it is updated with a new dataset that represents the new conditions.

The necessity of infilling for applying the model online is an issue which has been much investigated. There is no clear consensus on which infilling method is the best. García-Muñoz, Kourti, and MacGregor (2004) note that because the control charts are tailored to whichever infilling method is chosen, the choice of infilling method is not critical for control chart performance. In a comparison of many monitoring methods, Rato et al. (2016) observe that projection to model plane infilling performed slightly better in the two datasets they considered. Nomikos and MacGregor (1995b) themselves propose an alternative approach where infilling is not necessary: namely building a separate MPCA model for each time-point. They only resort to infilling due to the computational cost of building a large number of MPCA models. Ramaker et al. (2005) pursue this idea, referring to the method as a time-evolving MPCA model. They observe faster speed of detection compared to the traditional MPCA with infilling. Despite this, the classical MPCA model using some form of infilling still seems to be the more frequently used method in the literature.

One aspect of the method which is rarely discussed is that of using two control charts to monitor the process. This complicates the assessment of false alarm and detection rates. Furthermore, in the application of MPCA in Paper 3, the T^2 chart was found to be practically redundant as almost all out of control points in the T^2 chart were signalled out of control in the Q chart. For the T^2 chart to be useful, it must be plausible that faults may occur which vary *within* the R dimensional model plane only. It seems unlikely that this is the case, so it should be sufficient to only use the Q chart and thereby avoid an inflated false alarm rate.

2.2.3 Variable-wise unfolding and related methods

Wold et al. (1998) developed a monitoring method based on a different way of unfolding the three-way data structure of batch process data. This type of unfolding has become known as variable wise unfolding and is illustrated in Fig. 2.2. It entails arranging the data for each batch vertically such that a $IK \times J$ matrix, $\mathbf{X}$ is obtained. Each column of this matrix is centered and scaled. This means that the average variable trajectories are not subtracted as they are in the MPCA approach.

Next, a $IK \times 1$ maturity variable, $\mathbf{y}$ is constructed which is simply the batch time for each row in $\mathbf{X}$. A partial least squares (PLS) model is built to predict $\mathbf{y}$ from $\mathbf{X}$. PLS will be further discussed in Section 2.3, but here it is sufficient to note that it results in scores $\mathbf{T}$ ($IK \times R$) and loadings $\mathbf{P}$ ($J \times R$), in a similar way to PCA, with R being the number of retained components. The reason for obtaining scores and loadings via PLS prediction of a maturity variable, rather than by PCA directly appears to be to try and account for the non-stationary trajectories that are still present in this version of unfolding. Control charts for Hotelling's T^2 and Q can be constructed based on these scores and loadings in a similar manner to that described previously. An advantage of this approach is that it does not require any infilling of future values in order to apply the model online. At time k, the J-length vector of current variable values can be used directly. Also, batches of varying duration do not pose a challenge, as the data can still be unfolded variable-wise. A major disadvantage is that the same loading vectors are used for all time points of the batch, implying that the correlation structure should be the same at any time in a batch. This also means that the R scores of a batch are still non-stationary trajectories. For these reasons, Westerhuis, Kourti, and MacGregor (1999) state that variable wise unfolding is not appropriate for monitoring batch processes.

Lee, Yoo, and Lee (2004) try to remedy the shortcoming of monitoring via variable wise unfolding (not addressing the time-varying nature of variable trajectories), by first unfolding batch-wise, and centering and scaling the columns of the data so mean trajectories are subtracted. Then the data is re-shaped in the manner of variable wise unfolding before building the PCA model. However, simply subtracting the mean variable trajectories cannot adequately account for autocorrelation in the time dimension.

Chen and Liu (2002) develop a Dynamic PCA method for batch process monitoring. This entails unfolding the reference data variable wise, and then appending columns consisting of the J variables lagged in time. That is, for a chosen lag of d, a row of their unfolded matrix has Jd elements and consists of the J variables of a batch at time t, followed by the J variables at time $t-1$ and so on, ending with the J variables at time $t-d$. Then PCA is applied, and T^2 and Q charts used to monitor future batches. Thus, the loading vectors of the PCA model for this data take into account autocorrelations of the variables of lags up to d. The problem remains that the model does not adequately account for the non-stationary trajectories of batch processes. For example, with reference to the penicillin data in Fig. 2.1, it is unrealistic to treat the (auto-)correlation structure at time 300 the same as the (auto-)correlation structure at time 50, but this is a consequence of the Dynamic PCA approach. Dynamic PCA had previously been applied for monitoring continuous processes (Ku, Storer, and Georgakis 1995) where it is more appropriate.

2.2.4 Three-way methods

An alternative to unfolding batch process data into a two-way matrix so that standard bi-linear methods like PCA can be applied, is to use methods that can directly model the three-way structure of the original data. Boqué and Smilde (1999) and Louwerse and Smilde (2000) use PARAFAC and Tucker3 models for batch process monitoring. An illustration of how these models deconstruct the three-way batch process data is shown in Fig. 2.3. These methods are tri-linear extensions of classical PCA on bilinear data. Just as with PCA, R scores are obtained for each batch, as well as R loading vectors in the variable mode of the array. However, PARAFAC and Tucker3 also generate loading vectors in the time mode. In the Tucker3 model there is a core array ($\underline{\mathbf{H}}$ in Fig. 2.3) which allows a kind of interaction between the loadings in the variable mode and the loadings in the time mode. Thus, in these three-way methods the time-varying trajectories are dealt with by the time mode loadings. If the Tucker3 core array is a super-diagonal array, with 1's on the diagonal, then the Tucker3 model reduces to a PARAFAC model. For both methods, control charts for Hotellings's T^2 and Q can be constructed. Louwerse and Smilde (2000) achieve online monitoring by simply fitting a new model for different time periods in the process (the first model for times 0 to 20, a second model for times 0 to 40 and so on).

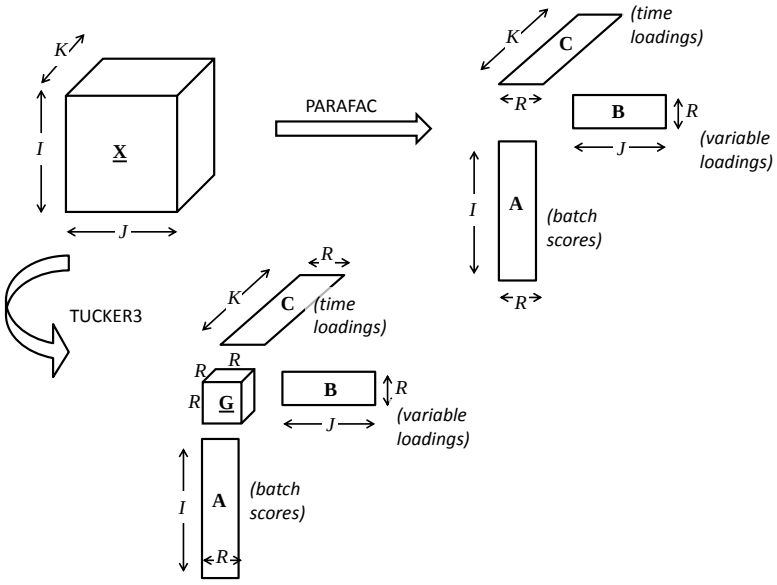

Figure 2.3: Illustration of array decomposition for PARAFAC and Tucker3 models.

The three-way models are more parsimonious than the MPCA model, with far fewer parameters. Supposing $R = 3$ components are retained in all models, and assuming $I = 50, J = 10, K = 200$ then the total number of fitted values contained

in the scores and loadings is:

$$\text{MPCA:} \quad IR + (JK)R = (I + JK)R = 6,150 \tag{2.10}$$

$$\text{Tucker3:} \quad IR + JR + KR + R^3 = (I + J + K + R^2)R = 807 \tag{2.11}$$

$$\text{PARAFAC:} \quad IR + JR + KR = (I + J + K)R = 780 \tag{2.12}$$

However, as noted by Westerhuis, Kourti, and MacGregor (1999) the PARAFAC decomposition is too restrictive for batch process data: It implies that the same time-trajectory shape described by each time-mode loading, is present in all J variables. This is clearly problematic, as in batch processes it is common for different variables to have radically different shapes. The Tucker3 approach shows more promise, as the core-matrix enables greater flexibility in modelling the three way geometry of batch process data.

In comparing the PARAFAC, Tucker3 and MPCA monitoring methods on a simulated dataset Louwerse and Smilde (2000) conclude that based on the T^2 chart, PARAFAC and Tucker3 perform slightly better than MPCA, whilst based on the Q chart, MPCA performs better. In their words, "there is no king method". In contrast, Westerhuis, Kourti, and MacGregor (1999) more decisively conclude that MPCA is the preferred method in comparison to PARAFAC and Tucker3. This is based on the monitoring performance of the methods on a real industrial dataset of batch polymerisation, and a simulated dataset, as well as on a discussion of the nature of the models and their applicability to batch processes. In a comprehensive comparison of many monitoring methods, Rato et al. (2016) observe that 2-way methods (MPCA) perform better than the 3-way methods, and 2-way method performance is more stable across different datasets. All proponents of 2-way methods refer to its greater flexibility to model the process.

In closing, of the three-way methods, the PARAFAC approach should be discarded for batch process monitoring, as its assumptions about the geometry of the data are unrealistic. The parsimony of the Tucker3 model is appealing. However, a rarely discussed disadvantage of Tucker3 is the difficulty of interpretation. With the Tucker3 model, the influence of the core array must be accounted for in order to correctly interpret the loadings in the time and variable modes, which is a challenging task. Despite the greater number of parameters of the MPCA model, the loadings can be directly interpreted. For these reasons, in addition to the findings cited above, it seems that MPCA is the preferred method for batch process monitoring.

2.2.5 Alignment of batch process data

The methods referred to so far all require the same number of observations in each batch, K. In reality, it is common for the duration of batches to vary, and therefore to have differing numbers of observations. This is especially the case in bio-processes,

where, for example, growth rates of cell cultures may vary, and so batch completion time is uncertain. When batch duration varies, it is necessary to standardise the number of observations per batch, so that the models can be fit to the reference dataset. A simple approach is to truncate all batches in the dataset to the length of the shortest batch. However, then the resulting model cannot be used to monitor periods of a batch after the length of the shortest reference batch. Besides, variation of overall batch duration is not the only type of time variation of concern. In addition, there may be variation in the timing of any of the sequence of events that take place during a particular process. For example, with reference to the penicillin dataset (Fig. 2.1), one batch may attain peak base-flow at time 40, whilst another batch may attain peak base-flow at time 50. An MPCA model fitted directly to the unfolded data, treats "base-flow at time 45" as a variable in itself. However, due to time variation, "base-flow at time 45" represents different events for different batches. It is like analysing the heights of a sample of people, and then discovering that for some people the height was measured to the top of the head, and in others to the shoulders. This is the main argument for synchronisation of batch process data.

2.2.5.1 Indicator Variable Approach

Nomikos and MacGregor (1995b) suggest aligning batch process data by using an indicator variable. This approach entails identifying a variable in the process data which has the same starting and ending value for all batches, and is monotonic. If the same quantity of some substance is gradually added to all batches, then the variable which represents this quantity makes an ideal indicator variable. Then the other variables are re-sampled at even increments of the chosen indicator variable. Thus, the progress of a batch is not measured in time, but in rate of consumption of the indicator variable. It is a simple and elegant solution to the alignment problem. However, the major drawback is that for many batch processes, a suitable indicator variable may not exist. In fact, a trouble-free alignment using a single indicator variable is hard to find in the literature. In practice, it is often necessary to use a combination of several variables as the indicator variable. García-Muñoz, Kourti, and MacGregor (2003) use three indicator variables to align an industrial drying batch process, and García-Muñoz et al. (2011) resort to using four different indicator variables in a pharmaceutical batch process. In both cases, the processes consist of several phases, and a different indicator variable is used for each phase in the process. The indicator variable method is worth considering, but is clearly not a general solution that can be used for all batch processes.

2.2.5.2 The Dynamic Time Warping Algorithm

Kassidas, MacGregor, and Taylor (1998) introduce a framework for aligning batch process data using dynamic time warping (DTW). DTW was originally developed in the field of speech recognition (Sakoe and Chiba 1978). In speech recognition, the problem is to match the audio signal of a spoken query word to the most similar entry

in a database of labelled audio signals. DTW provides a measure of the similarity of two time series, which is not sensitive to timing differences in the time series, and in so doing provides a way to align them. As DTW plays a central role in the contributions of this thesis, a summary of the DTW algorithm from Sakoe and Chiba (1978) is presented in this section, before returning to the literature on DTW applied to batch processes

Let $\mathbf{x}$ $(m \times 1)$ and $\mathbf{y}$ $(n \times 1)$ be two real-valued times series, referred to as the query trajectory and the reference trajectory, respectively. Then the goal of DTW is to find the optimal way of stretching and compressing $\mathbf{x}$ and $\mathbf{y}$ to make them resemble each other as closely as possible, and to quantify the resulting similarity of the warped trajectories.

The first step is to construct the local distance matrix, $\mathbf{C}$ $(m \times n)$. The $(i,j)^{th}$ element of $\mathbf{C}$ is simply the distance between the i^{th} element of $\mathbf{x}$ and the j^{th} element of $\mathbf{y}$ (denoted by x_i and y_j respectively). Often, the squared Euclidean distance is used

$$C_{i,j} = (x_i - y_j)^2 \tag{2.13}$$

DTW finds warping paths, f, through the local distance matrix $\mathbf{C}$ that represent mappings, between $\mathbf{x}$ and $\mathbf{y}$. $f = f_1, f_2, \ldots, f_T$ where $f_t = (i,j)_t$. The warping path is usually subject to the following constraints:

- Boundary conditions: $f_1 = (1,1)$ and $f_T = (m,n)$, i.e., the first elements of $\mathbf{x}$ and $\mathbf{y}$ should be aligned to each other as well as their last elements
- Continuity: If $f_t = (i,j)$ then $f_{t+1} = (i+a, j+b)$ where $a, b \leq 1$, i.e., the warping path cannot skip cells in the local distance matrix
- Monotonicity: If $f_t = (i,j)$ then $f_{t+1} = (i+a, j+b)$ where $a, b \geq 0$, i.e., each pairing cannot go backwards in time in either $\mathbf{x}$ or $\mathbf{y}$.

The accumulated distance between the two sequences under a warping, f, is given by

$$D_f(\mathbf{x}, \mathbf{y}) = \frac{1}{n+m} \sum_{t=1}^{T} \omega_t C_{f_t} \tag{2.14}$$

where $\omega_t = 2$ if warping the function from f_{t-1} to f_t takes a diagonal step and $\omega_t = 1$ otherwise. The step weights are used to compensate for warping paths of different lengths. DTW identifies the warping function which minimises the accumulated distance D_f and the corresponding value

$$D = \min_f [D_f(\mathbf{x}, \mathbf{y})] \tag{2.15}$$

is the DTW distance between the two sequences. Eq. 2.15 is solved using dynamic programming so it is not necessary to calculate every possible warping function. Dynamic programming solves a complex problem by breaking it down into many simpler sub-problems. In this case, the sub-problem can be framed as "suppose you are standing on the $(i,j)^{th}$ cell of $\mathbf{C}$, then, which preceding cell can you have stepped from, in order to have contributed the least possible distance to D_f?". The solution consists of constructing an accumulated cost matrix, $\mathbf{G}$ $(n \times m)$, the elements of which are filled in recursively, starting at $(1,1)$ and ending at (n,m) using the following relation

$$G_{i,j} = \min \begin{bmatrix} G_{i,j-1} + C_{i,j}, \\ G_{i-1,j-1} + 2C_{i,j}, \\ G_{i-1,j} + C_{i,j} \end{bmatrix} \tag{2.16}$$

where it is convention to assume $G_{0,0} = 0$ in order to fill in the first cell. (Note that the factor 2 in middle line of Eq. 2.16 corresponds to the step-weight ω_t for a diagonal step in Eq. 2.14). Once the accumulated cost matrix is completed, then (2.15) has been solved, and the DTW distance is $D = \frac{1}{m+n} G_{(m,n)}$. The optimal warping path, f, implied by the DTW distance can be identified using a straight forward back-tracking procedure, where, starting at (m,n), the predecessor point that solved Eq. 2.16 is identified.

Fig. 2.4 shows the components of the DTW algorithm for alignment of two short trajectories. The computation of $C_{4,3}$ and $G_{4,3}$ is shown. The optimal warping function for this alignment is $f = [(1,1),(1,2),(2,3),(3,3),(4,4),(5,4),(6,5)]$. Thus, under this warping function the trajectories become $\mathbf{x}_{\text{warped}} = (x_1, x_1, x_2, x_3, x_4, x_5, x_6)$ and $\mathbf{y}_{\text{warped}} = (y_1, y_2, y_3, y_3, y_4, y_4, y_5)$. The DTW distance is $D = \frac{1}{m+n} G_{(m,n)} = 1$

There are two ways in which the basic DTW algorithm just outlined is often adapted, both of which are designed to constrain the amount of permissible warping in the search space for the optimal warping function. The first is to use a global constraint, which constrains the warping function to a chosen sub-region of the accumulated distance matrix. A common global constraint is the band constraint, which, for band-width r, limits the warping function to a band of width $2r$, centered along the main diagonal of the accumulated distance matrix. The effect of a band constraint is that the maximum allowable time difference between the aligned $\mathbf{x}$ and $\mathbf{y}$ trajectories is r.

The second type of constraint is the local constraint. In the basic algorithm, there is no limit to the number of consecutive horizontal steps, or consecutive vertical steps in the warping path, which means that unlimited contractions and dilations of the trajectories are allowed. Sakoe and Chiba (1978) present a framework for local constraints consisting of allowable step-patterns. Each step-pattern is denoted by the parameter P which indicates how many diagonal steps must be enforced, before a horizontal or vertical step may take place in the warping function. In the basic algorithm, $P = 0$ (no diagonal steps are required before a horizontal or vertical step).

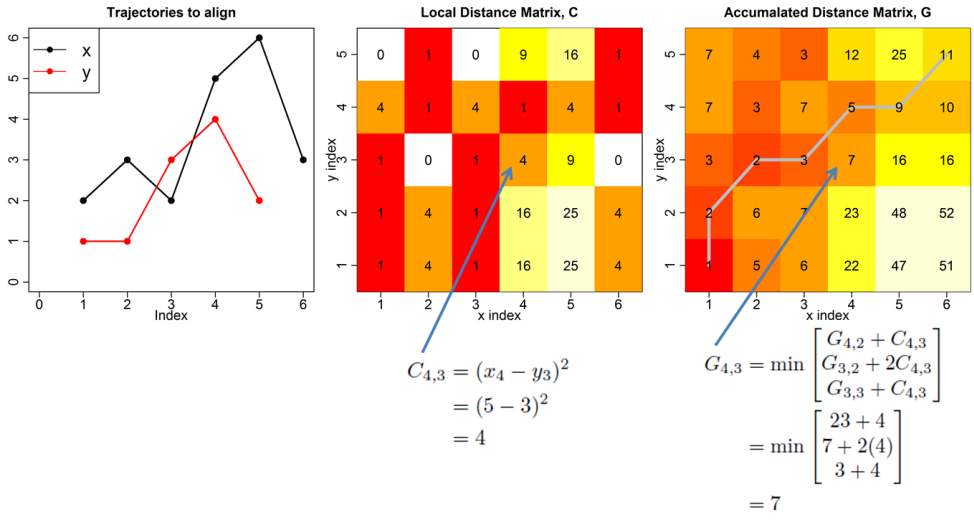

Figure 2.4: Small example of applying DTW to align two trajectories.

The step-patterns for $P = 0, 1/2, 1$ and 2 are shown in Fig. 2.5. The step-pattern for $P = 0$ corresponds to no constraint and is a graphical representation of Eq. 2.16 in the basic algorithm. Similar recursion formulas exist for the other step-patterns.

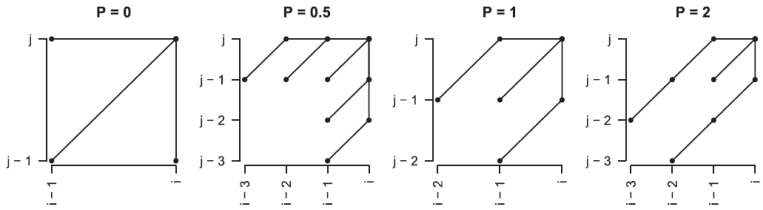

Figure 2.5: Local constraint step patterns.

Suppose $\mathbf{X}$ $(n \times J)$ and $\mathbf{Y}$ $(m \times J)$ are two multivariate time-series. Then the only adjustment required in order to use the DTW algorithm, is to define a multivariate local distance, and let $C_{i,j}$ be this distance between the i^{th} row of $\mathbf{X}$ and the j^{th} row of $\mathbf{Y}$. After $\mathbf{C}$ is calculated, the rest of the algorithm is the same as for univariate time-series. However, the curse of dimensionality must be considered, and Shokoohi-Yekta, Wang, and Keogh (2015) observe that in many applications, DTW with three or four variables included in the local distance outperforms DTW using greater numbers of variables.

2.2.5.3 Dynamic Time Warping for batch process data

Kassidas, MacGregor, and Taylor (1998) show how DTW could be used to synchronise batch process data, and thereby resolve the problem of batches of varying duration, and other timing differences. The method entails identifying a reference batch in the dataset, and aligning all other batches to this reference batch using DTW. It must be noted that an additional step is introduced in order to standardise the lengths of the warped batches. This is because DTW results in warped trajectories of length T (the length of the warping path), which can vary from batch to batch. The additional step consists of averaging those observations which are aligned to the same observation of the reference batch. Then the aligned batches all have the same number of observations as the reference batch. For example, suppose $\mathbf{y}$ in the example from Fig. 2.4 is the reference trajectory, then applying the averaging step results in a warped $\mathbf{x}$ trajectory of $\mathbf{x}_{\text{warped2}} = (x_1, x_1, \frac{x_2+x_3}{2}, \frac{x_4+x_5}{2}, x_6)$, because the warping function aligns x_2 and x_3 to y_3, and similarly x_4 and x_5 are both aligned to y_4. $\mathbf{x}_{\text{warped2}}$ then has the same length as $\mathbf{y}$.

Kassidas, MacGregor, and Taylor (1998) apply multivariate DTW based on all the variables in the process. This means it is necessary to scale the variables so that the alignment is not dominated by those variables with larger measurement units. For scaling, the authors divide each variable by its mean range over batches in the dataset. They define the multivariate local distance as

$$C_{i,j} = (\mathbf{X}_{i,.} - \mathbf{Y}_{j,.})' \mathbf{W} (\mathbf{X}_{i,.} - \mathbf{Y}_{j,.}) \tag{2.17}$$

where $\mathbf{X}_{i,.}$ is the vector formed by the i^{th} row of the query batch $\mathbf{X}$, and $\mathbf{Y}_{j,.}$ is the vector formed by the j^{th} row of the reference batch $\mathbf{Y}$ (both vectors are therefore $J \times 1$). $\mathbf{W}$ $(J \times J)$ is a diagonal matrix used to weight the different variables, with entry $W_{j,j}$ being the weight of variable j. Variables with a large weight will contribute more to the local distance and thereby have greater influence on the DTW solution. The idea is that some variables are more informative for alignment than others and should therefore have a larger weight. The authors present a computational method for determining $\mathbf{W}$. This method assigns greater weight to variables which are more consistent from batch to batch (have less batch to batch variation).

The procedure for incorporating DTW alignment into the MPCA monitoring framework is summarised as follows. First the reference batch is selected as the batch with duration closest to the median duration in the dataset. The variables are scaled, $\mathbf{W}$ is calculated, and DTW is used to align all batches to the reference batch. This results in a dataset where all batches now have the same number of observations, and the events within a batch are synchronised. It is important to retain the original time information of each batch. This is done by adding a new time variable to each aligned batch, which contains the original real-time of each observation. An MPCA model can then be fitted to the aligned data which now has dimensions I batches $\times$ $J+1$ variables (the original J variables and the "real-time" variable) $\times$ K

observations (the number of observations in the reference batch).

To monitor a new batch online, it is necessary to apply an online version of DTW which relaxes the end-point constraint and thereby allows the DTW algorithm to identify the leading portion of the reference batch which most closely resembles the part of the ongoing batch completed so far (Kassidas, MacGregor, and Taylor 1998). The online version of DTW is equivalent to performing K regular DTW alignments between the ongoing batch so far, and all possible leading sections of the reference batch, and selecting that leading section which results in the smallest DTW distance. For each new observation in the ongoing batch, a new online alignment is performed to synchronise the data so far, then the MPCA model is applied as usual.

Ramaker et al. (2003) propose a different method of calculating the variable weights, $\mathbf{W}$, for DTW alignment of batch data. This method is based on the degree to which the surface plot of the local distance matrix for each variable (i.e. the middle image of Fig. 2.4) has a valley along its diagonal. Variables with a more pronounced valley are assumed to be better guides for alignment and are given greater weight.

Other authors focus on the online implementation of DTW. A feature of the original proposal for online DTW is that it can result in radically different alignments for each new observation in the ongoing batch: at one moment the ongoing batch may be aligned to the first 31 minutes of the reference batch, then the next minute it may be aligned to the first 23 minutes of the reference batch. González-Martínez, Ferrer, and Westerhuis (2011) propose a method for increasing the stability of online DTW which involves only allowing revisions to the warping path within a window of the k most recent observations of the ongoing batch. However, it can be questioned whether it is a good idea to suppress the revisions in alignment. Perhaps it is natural that the alignments are revised in light of new information on the process. González-Martínez et al. (2014) develop further their approach to online DTW alignment, and provide a method for dealing with four different types of asynchronisms (one type being, for example, batches that start at different points in the process). The authors conclude that synchronisation is critical to monitoring performance.

The issue of constraining the degree of warping in DTW for batch process data is usually tackled by using global constraints. González-Martínez, Ferrer, and Westerhuis (2011) define global constraints based on an empirical envelop around the warping functions from offline alignment of the reference dataset. Lu et al. (2016) develop a new method for defining global constraints which aims to only allow warping in regions of the trajectories that do not contain important features. However, this implies that any timing differences between the new batch and reference batch only take place in periods without the selected features, which is problematic for a literal interpretation of the warping function as representing the progress of a batch relative to the reference batch.

2.2.5.4 Correlation Optimised Warping

In the chemometrics field, Correlation Optimised Warping (COW) was developed for correcting peak-shifts in chromatographic data (Nielsen, Carstensen, and Smedsgaard 1998). This method is in many ways very similar to DTW, and aims to identify the optimal warping function to warp a trajectory such that it most resembles a reference trajectory subject to some constraints. The main difference is that COW aims to maximise the correlation of the warped trajectories, rather than minimise the distance between them. In brief, under COW the query trajectory is divided into segments of equal length, L. Each segment is then either compressed or stretched, up to some maximum amount determined by a slack parameter, such that the correlation between the warped query and reference trajectories is maximised. A dynamic programming algorithm is used to find the optimum warping for a given slack value and segment length. Skov et al. (2007) present a method for automatic selection of slack value and segment length. COW was first applied in an industrial batch process setting by Fransson and Folestad 2006, who present a way to use COW in online monitoring of a drying process. One difference between DTW and COW is that COW cannot directly handle multivariate time series, whereas the multivariate extension of DTW is straightforward. This may be why COW appears to be preferred in the chemometrics domain to align, for example, univariate chromatographic data, whilst in statistical analysis of batch processes DTW is more commonly used.

2.2.5.5 Models with in-built alignment

The alignment methods discussed above can all be used to align the data, prior to analysis with any of the monitoring methods, including the three-way methods. Alternatively to performing alignment as a pre-processing step, there exist variants of some of the three-way models which do not require pre-alignment, and which can account for mis-aligned data within the model structure. Wise, Gallagher, and Martin (2001) apply a PARAFAC2 model for batch process monitoring. In a PARAFAC2 model, the time loading vectors can change from batch to batch, so batches of varying duration can be handled directly. Similarly, Luo et al. (2014) develop a GTucker2 (Generalized Tucker2) model for batch process monitoring. Although these methods are more flexible than their parent methods, they still have the disadvantage that the same time profiles in the models must be applied to all variables, even though trajectory shapes may be radically different for different variables.

2.2.6 Closing remarks on monitoring

In Section 2.2, an overview of methods found in the literature for monitoring batch processes has been given. The focus has been on latent structure methods and the alignment problem. It must be noted that all the latent structure methods discussed here are linear, in that they assume the process data from normal operating conditions can be represented by a linear combination of R scores and loadings, where R is much

smaller than the original variable dimension. Besides the latent structure methods, many of the more recent machine learning algorithms have been applied to monitoring batch process data in some form or other. Indeed, Rato et al. (2016) and Rato et al. (2018) remark that the number of new methods for batch process monitoring appearing in the literature seems to be accelerating, which is why they propose a number of guidelines for assessment of the methods. They make a case for greater use of Receiver Operating Characteristic (ROC) curves to assess performance, to avoid unfair comparison of methods that are tuned to different false alarm rates.

Van Impe and Gins (2015) highlight another issue related to the proliferation of batch process monitoring methods, in that the majority of authors demonstrate their methods on their own simulated datasets, often including only two or three faulty batches. Therefore, Van Impe and Gins generate a simulated dataset covering an extensive variety of faults and normal operating conditions which they hope may be used as a benchmark in future batch process monitoring literature. These recent efforts reflect a need for further comparison and assessment of the many approaches to monitoring of batch processes.

2.3 Prediction

Batch process monitoring is usually an "unsupervised" problem - the goal is to identify whether a new batch is in some way out of the ordinary compared to previously observed batches, without any prior classification of what makes an ordinary batch. In contrast, batch process prediction is a supervised problem. Typically, a dataset of process data and response data (quality parameters) is available for a number of batches, and the problem is to fit a model which can predict the response data based on the process data. This model can then be used to predict the response data for future batches, saving the time and expense of having to directly measure the response variable in the future. If the model allows, predictions can be made online as a batch is progressing, in which case it is often referred to as a "soft sensor" as real-time values for the response variables are obtained using software, rather than being measured directly by physical hardware.

Prediction from batch-wise unfolded data is usually a case of $p \gg n$. ("p is much greater than n", where p is the number of variables and n the number of observations). In the present case, $p = JK$ and $n = I$. Most traditional statistical methods, such as ordinary least squares regression, were developed under the assumption that $p < n$, and are not well suited to the $p \gg n$ case for several reasons. Firstly, the $p \gg n$ case is ill-posed so a unique solution to solving the variable coefficients that minimise the sum of squared errors does not exist. Secondly, when there are a large number of variables, there is frequently a high degree of collinearity between many variables resulting in unstable coefficient estimation. Two common approaches to the $p \gg n$ problem are to apply a latent structure method like PLS, so the effective number of

variables is reduced, or to impose additional regularisation constraints on ordinary least squares regression to obtain more reliable coefficient estimates.

2.3.1 Partial Least Squares based methods

The classical approach to batch process prediction was formalised by Nomikos and MacGregor (1995a) and referred to as Multiway Partial Least Squares (MPLS). It is a natural counterpart to their MPCA monitoring method, where in addition to the three-way process data $\underline{\mathbf{X}}$ $(I \times J \times K)$, there is now also a set of L final quality parameters for each batch, forming the response matrix $\mathbf{Y}$ $(I \times L)$. The problem of predicting $\mathbf{Y}$ based on the process data, $\underline{\mathbf{X}}$, is approached by unfolding $\underline{\mathbf{X}}$ batch-wise, and centering and scaling each column to obtain $\mathbf{X}$ $(I \times JK)$ just as for MPCA (Nomikos and MacGregor 1994). The columns of $\mathbf{Y}$ are also usually centered and scaled. Then the classic Partial Least Squares (PLS) method (Geladi and Kowalski 1986) can be applied to predict $\mathbf{Y}$ from $\mathbf{X}$. This latent-structure regression method de-constructs both $\mathbf{X}$ and $\mathbf{Y}$ into scores and loadings in such a way that the covariance between the $\mathbf{X}$-scores and $\mathbf{Y}$-scores is maximised within the following outer relationship (see Table 2.1 for definitions of the notation used):

$$\mathbf{X} = \mathbf{T}\mathbf{P}' + \mathbf{E} \text{ and } \mathbf{Y} = \mathbf{U}\mathbf{Q}' + \mathbf{F}^* \tag{2.18}$$

$\mathbf{Y}$ is predicted by the scores of $\mathbf{X}$ through the inner relationship:

$$\mathbf{Y} = \mathbf{T}\mathbf{B}\mathbf{Q}' + \mathbf{F} \tag{2.19}$$

where the PLS algorithm calculates the scores and loadings and minimises $||\mathbf{F}||$ using alternating least squares.

The number of latent variables to keep in the PLS model, R, can be selected using cross-validation by minimising the cross-validation sum of squared errors of the response variables.

To apply the model online during a batch, and thus obtain online predictions of final batch quality parameters, one must use an infilling method, just as for online application of MPCA. Other authors have avoided the infilling step by building time-evolving MPLS models (Gunther, Conner, and Seborg 2009). An advantage of the MPLS approach for prediction is that it incorporates a description of the process data, and the familiar Hotelling's T^2 and Q charts can be constructed for monitoring the $\mathbf{X}$-scores online. Therefore, both monitoring and prediction are addressed within the same framework.

A drawback of MPLS as a prediction method, is the difficulty of interpretation of the model, and of understanding what factors in the process data really influence the quality parameters, which is of great commercial interest. Every variable at every time-point is included in the model, and understanding the model requires

Table 2.1: PLS Notation.

Symbol	Definition
$\mathbf{X}$	$(I \times JK)$ Unfolded process data
$\mathbf{Y}$	$(I \times L)$ Response variables
$\mathbf{T}$	$(I \times R)$ $\mathbf{X}$-scores
$\mathbf{U}$	$(I \times R)$ $\mathbf{Y}$-scores
$\mathbf{P}$	$(JK \times R)$ $\mathbf{X}$-loadings
$\mathbf{Q}$	$(L \times R)$ $\mathbf{Y}$-loadings
$\mathbf{E}$	$(I \times JK)$ $\mathbf{X}$-errors
$\mathbf{F}^*$	$(I \times L)$ $\mathbf{Y}$-errors in outer relation
$\mathbf{F}$	$(I \times L)$ $\mathbf{Y}$-errors in inner relation
$\mathbf{B}$	$(R \times R)$ Coefficients relating scores of $\mathbf{X}$ to scores of $\mathbf{y}$
R	Number of latent variables retained in PLS model

examination of the loading vectors. Techniques for estimating the relative importance of variables in the PLS model have been investigated (Chong and Jun 2005), and can be used to exclude variables and refit the model. Lu and Gao (2005) present a strategy for identifying the specific time periods, and variables within each time period, that are critical to quality, to use in online quality prediction with PLS.

PLS combined with variable-wise unfolding has also been investigated (**e.g**; Ündey, Ertunç, and Çinar 2003), but again, variable-wise unfolding contradicts the fundamental assumption about batch processes that they have a dynamic structure with changing relationships between variables over time within a batch.

2.3.2 Other regression based methods

The general formulation of a regression model is

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{2.20}$$

For simplicity, it is now assumed there is a single response variable to be estimated, $\mathbf{y}$ $(I \times 1)$. Again, $\mathbf{X}$ $(I \times JK)$ is the batch-wise unfolded process data. The regression coefficients are $\boldsymbol{\beta}$ $(JK \times 1)$, and $\boldsymbol{\epsilon}$ are the errors. In ordinary least squares, $\boldsymbol{\beta}$ is calculated by minimising the sum of squared errors:

$$\hat{\boldsymbol{\beta}}_{OLS} = \arg\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\mathbf{y} \tag{2.21}$$

where $||\cdot||_2$ is the L2-norm. However, when $JK > I$, then $\mathbf{X}'\mathbf{X}$ is singular and so cannot be inverted. Hoerl and Kennard (1970) propose overcoming this problem by

adding a small constant, λ, to the diagonal of $\mathbf{X}'\mathbf{X}$ in Eq. 2.21, giving rise to a technique known as ridge regression. This regularisation approach is equivalent to minimising both the sum of squared errors, *and* the L2-norm of the coefficient values:

$$\hat{\boldsymbol{\beta}}_{ridge} = \arg\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_2^2 = (\mathbf{X}'\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}'\mathbf{y} \tag{2.22}$$

where $\mathbf{I}$ ($JK \times JK$) is the identity matrix. Besides making the regression equation solvable, ridge regression has some advantages to OLS in cases where there is high co-linearity between variables. Then, by introducing a small bias in estimating the coefficients, ridge regression can lead to a smaller prediction error compared to OLS regression, a concept known as the bias-variance tradeoff (Hastie, Tibshirani, and Friedman 2001). The value of λ is typically selected by cross-validation. Although, batch process data contains many highly correlated variables, especially when batch-wise unfolded (consider, e.g., temperature at time 50 and temperature at time 51), ridge regression has not been much applied in the field. This may be because it is thought to perform very similarly to the more established PLS regression (Frank and Friedman 1993; Hastie, Tibshirani, and Friedman 2001) (PLS can be expressed in the general regression form of Eq. 2.20, with $\boldsymbol{\beta}_{PLS} = \mathbf{PBQ}'$).

An alternative regularisation approach proposed by Tibshirani (1996) entails replacing the L2-norm of $\boldsymbol{\beta}$ in Eq. 2.22, with the L1-norm. This approach was named lasso (least absolute shrinkage and selection operator) regression. The lasso coefficients are given by solving

$$\hat{\boldsymbol{\beta}}_{lasso} = \arg\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_1 \tag{2.23}$$

where $||\cdot||_1$ is the L1-norm. There is no closed form solution to Eq. 2.23, so it must be solved computationally using quadratic programming. The advantage of lasso regression is that as the value of λ increases, then more coefficients within $\boldsymbol{\beta}$ become exactly zero, and so the method performs automatic variable selection. This is useful for high-dimensional problems like batch-process prediction, as only a small subset of process variables and time periods are retained in the model, so the model is easier to interpret and can contribute to greater process understanding. Yan et al. (2014) adopt the lasso for predicting product quality in a batch process. Again, the tuning parameter λ is usually selected using cross validation.

One argument against lasso regression is that when variables are highly correlated, it will tend to arbitrarily select one out of a group of correlated variables and set coefficients for the others to zero. This behaviour can be limited by combining the lasso and ridge regularisation methods leading to the so-called elastic net regression (Zou and Hastie 2005). The elastic net coefficient estimate is defined as

$$\hat{\boldsymbol{\beta}}_{EN} = \arg\min_{\boldsymbol{\beta}} ||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda\big[\alpha||\boldsymbol{\beta}||_2^2 + (1-\alpha)||\boldsymbol{\beta}||_1\big] \tag{2.24}$$

The elastic net encourages a grouping effect in the variable selection, where strongly correlated groups of variables are either all included, or excluded, from the model

together, depending on the value of λ. Chiu and Yao (2013) apply elastic net regression to batch process prediction. Of course the disadvantage of elastic net over lasso regression is that there is an extra parameter to tune, requiring a more complex cross validation procedure.

2.3.3 Non-linear methods

All of the prediction methods discussed so far are strictly linear. Assuming that $\mathbf{X}$ is the batch-wise unfolded process data, the methods do not even consider two-way interactions. This is quite a big limitation, as the models assume that the response variable can be modelled as a simple linear combination of the input data. Despite this, no publications were found which attempted to incorporate two-way interactions in any of the linear methods above, for batch process prediction. However, many non-linear techniques have been applied to batch processes. For example, Desai et al. (2006) use support vector regression to develop a soft sensor for a fermentation batch process. Onel et al. (2018) adopt support vector machine based method for detecting faults in a simulated penicillin batch process. Desai et al. (2005) apply artificial neural networks to predict yield of a batch process. These non-linear approaches may be more capable of detecting the complex dynamics present in biological batch processes. The disadvantage is that they are "black box" models, so gaining new process understanding from the models is a challenge.

2.4 Chapter conclusion

This chapter has provided a sketch of the research landscape as it relates to the statistical analysis of bio-based processes. There is no shortage of methods for monitoring and quality prediction of batch processes. However, real experiences of applying the methods in industrial applications are poorly documented, and will hopefully be more forthcoming in the future. The focus of this chapter has been on those areas which have led to the contributions of the thesis. These contributions are presented in the following chapters.

CHAPTER 3

Aligning batch data with Chr. Hansen

Figure 3.1: Production facilities at Chr. Hansen, Avedøre (source: https://www.chr-hansen.com/en/media/image-library).

Early in the PhD project, a collaboration was formed with Chr. Hansen, a global bioscience company, whose main business area is producing bacteria cultures and enzymes for the food industry. The bacteria cultures are mainly produced for the dairy industry, but also for wine and meat products, amongst others. In fact, around 50 % of the cheese in the world contains at least one product from Chr. Hansen (Chr. Hansen A/S 2018). The company has several production sites world-wide, and the collaboration was with the site in Avedøre in Greater Copenhagen (Fig. 3.1). Here, the project focused on the data measured during batch production of bacteria cultures. Before defining project goals, time was spent gaining understanding of the production process, available data and needs of the company.

Chr. Hansen produces a large variety of bacteria cultures, which have been developed from their catalogue of almost 30,000 bacteria strains. Both single strain and multi-strain cultures are produced. These speciality products are well suited to batch production, as discussed in Chapter 2. For example, a batch can be produced for an Italian salami maker one day, and the next day a batch may be produced for a

Danish yoghurt dairy, according to market conditions.

The production line for making bacteria cultures has several steps. The project focused on the fermentation step. This step begins when the reactor vessel (fermenter) is charged with a small, concentrated quantity of the selected bacteria strain(s). The fermenter has been pre-filled with growth media consisting of a mixture of water, sugars and other nutrients. Conditions within the fermenter, such as pressure and temperature, are carefully controlled. The bacteria then consume the growth media and multiply, producing acid as a waste product in the process, and thereby gradually lowering the pH inside the fermenter. Eventually (roughly half-way through a batch), the pH drops to a set-point value (pre-defined by the company for each product), which triggers the second phase of the batch during which pH is automatically controlled by the addition of a base substance. Thus, during the second phase pH is kept at a constant level by the control mechanism, and bacteria activity is then reflected in the rate of base addition, rather than changes in pH in the fermenter. Finally, a process engineer decides when the batch should be stopped, based on specifications of the product, and thereby initiates rapid cooling of the fermenter.

The available process data consisted of variables measured by sensors in the fermenter during batch runs, including pH, temperature, pressure, level and base addition. A number of end of batch quality measurements were also made available. The general objective of the project was defined as determining how to use this data to understand what makes a ”good” batch. This could be by either monitoring of process data, thereby detecting if a fault occurs that would have an impact on product quality, or by predicting product quality parameters based on process data. In pursuing this goal, a number of challenges were encountered.

At Chr. Hansen, data is managed using a data historian. Data historians have become widespread in process industries to manage the increasing quantities of generated data. Due to prioritising disk space and retrieval speed, data historians perform compression on the raw sensor data before storing it (Thornhill, Shoukat Choudhury, and Shah 2004). Usually this compression is based on only retaining a sensor reading (and its time-stamp) when it differs by at least some amount (the sensor dead-band) compared to the previously stored reading. It is recommended that the deadband should be slightly narrower than the precision of the sensor. The result of data compression is that the different variables are stored at different sampling rates, and at unevenly spaced points in time throughout the batch. (it should be noted that already at the measurement stage, the sensors are likely to have different sampling rates, and the effect of data compression upon storage is in addition to this). Thus, the process data obtained from Chr. Hansen consisted of variables measured at different rates. This aspect of industrial data has received very little attention in the literature on process analysis, where almost all authors assume without comment that the measurements for all variables exist at the same regularly spaced sampling-points. However, Kourti (2003) and Klimkiewicz (2016) discuss some aspects of data compression for

modelling industrial data. In the present case, the data was retrieved from the historian in its stored format, after which each variable was linearly interpolated to obtain measurements at the same regularly spaced points.

Another challenge encountered in the project was the size of the data. Chr. Hansen produces a large variety of products and many batches are produced each day. However, the data for different products has very different properties, depending on the particular bacteria strains used. The durations of batches for one product can be half the duration of batches from another product. The features in the pH and base addition trajectories from a multi-strain culture are very different than those from a single strain culture. Therefore, it was appropriate to analyse products separately. This meant that obtaining a dataset of a large number of batches of the same product was difficult, as it may take a year or more before, for example, 50 batches of one product have been produced. As discussed in Chapter 2, batch process production is known to be suited for speciality products, but the fact the speciality products are produced in large variety and small numbers seems to be overlooked by most of the literature on batch process monitoring and prediction, where a large dataset of comparable batches is assumed. This aspect may be particularly relevant in bio-industry as opposed to, for example, manufacturing.

The difficulty of comparing products was compounded by issues related to the equipment used. Chr. Hansen is a global company with several production sites in different countries. Naturally, the same equipment is not used at every site. For example, there is variation in the size of the fermenters and in the control mechanisms for pH control. It was found that only batches of the same product, and produced in the same fermenter, were statistically comparable. This meant that the ambitious goal of obtaining a unified plant-wide, product-wide solution to determining good and bad batches had to be revised, to more product and equipment specific solutions. Then, by focusing on the more frequently produced products it was possible to obtain reasonable numbers of comparable batches.

Last of all, a challenge regarding the availability of ”bad” batches is worth noting. Much of the literature considers a dataset of both good and bad batches. Then it is possible to fit a model using good batches, and test whether it succeeds in detecting the bad batches. However, in real industrial settings, ”bad” batches are exceedingly rare as discarding of a batch is so costly that everything possible is done to avoid producing a bad batch. This might suggest that there is no need for statistical monitoring or prediction methods, as industry seems to be managing fine without them. However, the point to note is that without such methods, avoiding faults rests on the expertise and vigilance of the process engineers. Automatic monitoring and quality prediction can save resources and provide valuable assistance to the human experts. The challenge remains that without bad batches, validation of models is very difficult, and it is hard to understand what types of faults might occur so that models might be adapted to them.

After much preliminary data analysis and discussion of the production process with managers and engineers at Chr. Hansen, a direction for research was formed. For batches of the same product, substantial variation was observed in the rate of growth of the bacteria cultures (reflected in the trajectories of pH and base flow), and alignment of the process data was found to be highly relevant. Dynamic time warping was investigated to explain this time variation in the process. Current methods for constraining the algorithm in order to ensure realistic warping of the time dimension were found to be lacking, and therefore the use of local constraints was investigated and a method for selecting a local constraint was developed. The results of this research have been published and the article is included in the following section.

3.1 Paper 1: Selecting local constraint for alignment of batch process data with dynamic time warping

Spooner, M., D. Kold, and M. Kulahci. 2017. "Selecting local constraint for alignment of batch process data with dynamic time warping". *Chemometrics and Intelligent Laboratory Systems* 167:161–170.
https://doi.org/10.1016/j.chemolab.2017.05.019.

Chemometrics and Intelligent Laboratory Systems 167 (2017) 161–170

Contents lists available at ScienceDirect

Chemometrics and Intelligent Laboratory Systems

journal homepage: www.elsevier.com/locate/chemometrics

Selecting local constraint for alignment of batch process data with dynamic time warping

Max Spooner [a,*], David Kold [b], Murat Kulahci [a,c]

[a] DTU Compute, Technical University of Denmark, Kgs. Lyngby, Denmark
[b] Chr. Hansen A/S, Hvidovre, Denmark
[c] Department of Business Administration, Technology and Social Sciences, Luleå University of Technology, Luleå, Sweden

ARTICLE INFO

Keywords:
Batch process
Dynamic time warping
Local constraint
Global constraint
Step pattern
Trajectory synchronisation

ABSTRACT

There are two key reasons for aligning batch process data. The first is to obtain same-length batches so that standard methods of analysis may be applied, whilst the second reason is to synchronise events that take place during each batch so that the same event is associated with the same observation number for every batch. Dynamic time warping has been shown to be an effective method for meeting these objectives. This is based on a dynamic programming algorithm that aligns a batch to a reference batch, by stretching and compressing its local time dimension. The resulting "warping function" may be interpreted as a progress signature of the batch which may be appended to the aligned data for further analysis. For the warping function to be a realistic reflection of the progress of a batch, it is necessary to impose some constraints on the dynamic time warping algorithm, to avoid an alignment which is too aggressive and which contains pathological warping. Previous work has focused on addressing this issue using global constraints. In this work, we investigate the use of local constraints in dynamic time warping and define criteria for evaluating the degree of time distortion and variable synchronisation obtained. A local constraint scheme is extended to include constraints not previously considered, and a novel method for selecting the optimal local constraint with respect to the two criteria is proposed. For illustration, the method is applied to real data from an industrial bacteria fermentation process.

1. Introduction

In industrial batch processes data is often collected for I batches and J variables measured over the duration of the batch at K_i observations. It is often the case that the duration (number of observations, K_i) varies from batch to batch. However, most statistical methods for analysis of such three-way data require data in the format of an $I \times J \times K$ cube such that each batch has the same number of observations, K. This is the case in many methods of statistical batch process monitoring such as multi-way principal component analysis [13], multi-way partial least squares [14], PARAFAC/Tucker3 models [7] as well as numerous variations on these approaches found in the literature.

Furthermore, it is usually reasonable to suppose that the variation in overall batch duration is just one aspect of a more prevalent time-variation throughout the entirety of each batch. Industrial batch processes often consist of a complex sequence of stages. For example in the case of bacteria fermentation, the living cells multiply, consume nutrients, change food source and so on. The different stages may be reflected in the features (peaks, valleys, slopes etc.) of the variable trajectories. Each stage may occur faster or slower from batch to batch. Even if overall duration is standardised across batches, at similar time-points, different events may be occurring in each batch. Early approaches to processing of batch data did not address this aspect when dealing with uneven batch lengths. For example, all batches were simply cut to the same length, or linearly interpolated to obtain the same number of observations and form the $I \times J \times K$ data cube for further analysis. The first method ignores variation in local batch time, whilst the second method assumes that local batch time is expanded or contracted uniformly throughout the batch, which is unlikely in most processes and so the data will not be synchronised.

Reasons for synchronising batch data are well expressed in [3]. Firstly, if the aim is to apply a latent structure model, then such a model will encounter difficulty in identifying meaningful structure in the time dimension when the data is unsynchronised. This is because data for the same observation number, say k, will be contrasted for different batches and this does not make sense if observation number k corresponds to

* Corresponding author. DTU Compute, Asmussens Alle 322, 2800, Kgs. Lyngby, Denmark.
E-mail address: mpsp@dtu.dk (M. Spooner).

http://dx.doi.org/10.1016/j.chemolab.2017.05.019
Received 3 February 2017; Received in revised form 19 May 2017; Accepted 23 May 2017
Available online 25 May 2017

different events in different batches (e.g. in one batch observation 10 could occur during a lag phase, whilst in another batch observation 10 could occur during a growth phase). Synchronising the data will allow the model to better explain the variation in data for the same event across batches. A second reason for synchronising batch data is to gain a more detailed insight into the dynamics of the process.

A method which takes into account time variation throughout the batch as well as overall duration, has become known as the indicator variable method [2,11]. This method is useful if there exists a suitable indicator variable, which must be monotonically increasing or decreasing throughout each batch. Then the data can be interpolated against even increments of this variable which is used to indicate the progress of the batch or local batch time. However, for many batch processes there may be no suitable variable, or at best one that can only be used for one stage of the process such as in [2], in which case the advantage of this method, i.e. its simplicity, is lost.

Dynamic time warping is a synchronisation method which can align features at every point of each trajectory and does not require an indicator variable. This method originates from the field of speech recognition [17] and was later applied to batch process data by Kassidas et al. [6]. DTW aligns a query trajectory to a reference trajectory by finding the optimum matching of indices of each trajectory, subject to certain constraints, such that the overall distance between the trajectories is minimised. This optimum matching is found through a dynamic programming algorithm. The resulting time warping function contains information on how the local batch times of the trajectories progress. The application of DTW to batch process data has been investigated further by, for example, Ramaker et al. [16] and González-Martínez et al. [5].

An alternative alignment algorithm, Correlation Optimized Warping (COW), was presented by Nielsen et al. [12] for aligning chromatographic data. Tomasi et al. [18] evaluate both COW and DTW performance in the application of chromatographic data alignment and conclude that COW is preferable. With regards to alignment of batch process data (understood as engineering variables measured over time), COW has been further explored by [1], but DTW is the more widely used method in the literature. In this paper the practical issues of the application of DTW to industrial batch data will be discussed.

There are an array of options for tuning the DTW algorithm to a particular alignment problem and two of the key aspects are choice of global constraint and choice of local constraint. Briefly, the global constraint limits the maximum timing difference in the matching between query and reference trajectories found by DTW. There is freedom in how the trajectories are warped as long as the maximum timing difference is not exceeded. In contrast, the local constraint limits the amount of expansion/contraction that DTW applies throughout every point of the trajectories. If DTW is applied without either type of constraint, then the alignment may result in unrealistic, extreme warping. Researchers who have considered the problem of avoiding extreme distortions when aligning batch process data with DTW have focused on the global constraint, with little or no discussion of the local constraint [5]. Recently, Lu et al. [8] presented a method for specifying global constraints so that warping only takes place in selected regions of the trajectories such that key features are not distorted unduly. Again, a local constraint is not used. The primary goal of the present work is to provide a method for avoiding unrealistic warping with DTW whilst achieving a reasonable synchronisation of the key events in the trajectories.

In this paper, besides reviewing the many relevant aspects of DTW as applied to batch process data, we focus especially on the effects of local constraints. We argue that local constraints are preferable to global constraints for limiting unrealistic warping with DTW. The local constraint scheme of Sakoe and Chiba [17] is extended and stronger local constraints than previously seen in the literature are applied. A novel method for selecting the most appropriate local constraint is proposed which is readily generalisable to any batch process. A case study of real data from an industrial bacteria fermentation process is presented and the proposed methods are demonstrated on this data. Our local constraint method is shown to be superior to a global constraint in this case and the results are discussed.

2. Methods

2.1. Basic theory of DTW

In this section the various elements of DTW are introduced. Let $\mathbf{X} \in \mathbb{R}^{K_1 \times J}$ be a query batch of J variables collected at K_1 time points, and let $\mathbf{R} \in \mathbb{R}^{K_2 \times J}$ be a reference batch of J variables collected at K_2 time points. In order to align $\mathbf{X}$ to $\mathbf{R}$ with DTW, the first step is to construct a local distance matrix, $\mathbf{C} \in \mathbb{R}^{K_1 \times K_2}$ where $\mathbf{C}_{k_1,k_2}$ is the distance between observation k_1 in $\mathbf{X}$ and observation k_2 in $\mathbf{R}$ (denoted by the row vectors $\mathbf{X}_{k_1\bullet}$ and $\mathbf{R}_{k_2\bullet}$ respectively). Usually, the squared Euclidean distance is used

$$C_{k_1,k_2} = [\mathbf{X}_{k_1\bullet} - \mathbf{R}_{k_2\bullet}]\mathbf{W}[\mathbf{X}_{k_1\bullet} - \mathbf{R}_{k_2\bullet}]^T \tag{1}$$

where $\mathbf{W} \in \mathbb{R}^{J \times J}$ is a diagonal matrix with W_{jj} being the weight for the j^{th} variable. Variables with a large weight will contribute more to the distance and will therefore have greater influence on the alignment. The choice of these weights will be revisited in section 2.4. An advantage of using the squared Euclidean distance, rather than the Euclidean distance, is that the weights of each variable are preserved in the computed local distance.

The DTW algorithm considers a warping function $f(t)$, $t = 1, \ldots, T$

$$f(t) = (f_X(t), f_R(t)) \tag{2}$$

$$f_X(t) \in \{1, \ldots, K_1\} \tag{3}$$

$$f_R(t) \in \{1, \ldots, K_2\} \tag{4}$$

Thus, f maps the indices of $\mathbf{X}$ and $\mathbf{R}$ to a common time axis and defines a path through the cost-distance matrix consisting of T steps. After applying a given warping function, f, to $\mathbf{X}$ and $\mathbf{R}$ then the accumulated distance between the resulting warped trajectories is given by

$$D_f(\mathbf{X}, \mathbf{R}) = \frac{1}{M_f} \sum_{t=1}^{T} m_f(t) \mathbf{C}_{f(t)} \tag{5}$$

The term $m_f(t)$ denotes the weight assigned to the t^{th} step in the path, with M_f being the normalisation constant $\sum_{t=1}^{T} m_f(t)$.

Several basic constraints are usually imposed on the warping function f. In the case of global DTW alignment (synchronising all of $\mathbf{X}$ to all of $\mathbf{R}$), the following boundary constraints must apply:

$$f(1) = (1, 1) \text{ and } f(T) = (K_1, K_2) \tag{6}$$

The warping function must preserve the monotonicity of the original time dimension:

$$f_X(t) \leq f_X(t+1) \text{ and } f_R(t) \leq f_R(t+1) \tag{7}$$

Finally, the time warping function should be continuous:

$$f_X(t+1) - f_X(t) \leq 1 \text{ and } f_R(t+1) - f_R(t) \leq 1 \tag{8}$$

The goal of DTW is to identify warping function f satisfying these constraints, which minimises the accumulated distance $D_f(\mathbf{X}, \mathbf{R})$. The DTW algorithm achieves this through dynamic programming in $O(K_1 \cdot K_2)$ time. The details of this solution may be found in [10]. DTW thus produces a warping function $f = \text{argmin}_f(D_f(\mathbf{X}, \mathbf{R}))$ denoting how $\mathbf{X}$ and $\mathbf{R}$ can be warped to make them most similar, as well as the accumulated distance $D = \min_f(D_f(\mathbf{X}, \mathbf{R}))$ which quantifies the similarity between $\mathbf{X}$ and $\mathbf{R}$ after warping.

In order for the minimisation of the distance in Eq. (5) to be solved through dynamic programming it is necessary that the normalization constant (M_f) be independent of the path, f. This limits the possibilities

for the choice of step weights. The most well known step weights for symmetric DTW are those devised in [17] where diagonal steps are given a weight of 2, and horizontal and vertical steps are each given weights of 1. In this case, the normalization factor will be $M_f = K_1 + K_2$, independently of the warping path. An implication of this weighting scheme is that the minimisation of Eq. (5) does not favour warping paths of shorter length (small T). For the computed DTW distances to be comparable for different batches having different durations we suggest using this approach for the step weights.

Under the above constraints DTW operates symmetrically, warping both the reference and the query to obtain the closest alignment. The usual approach with batch process data is to apply a symmetric form of DTW between each batch $\mathbf{X}^{(i)}$ and the reference batch $\mathbf{R}$. However, the resulting warping functions will still have varying lengths. A second step is necessary to obtain warped batches of equal length to the reference batch. This step entails aggregating observations of the query batch whenever the warping function aligns several observations of the query to a single observation of the reference. This aggregation may be performed by taking the mean [6] or the median [8]. For example, in Fig. 1, the 3^{rd} observation of R is aligned to both observations 1 and 2 of X. Therefore, the 3^{rd} value of the warped X is taken to be the mean of X_1 and X_2. We choose to take the mean so that no information from the query trajectory is removed completely.

The basic form of dynamic time warping outlined in this section may be used to process batch data so each batch has the same length as the reference batch. However, the method will often result in extreme warpings and it is therefore preferable to impose additional constraints on the algorithm. In the following section we propose a novel method for selecting a local constraint so that unrealistic warping is avoided.

2.2. Local constraint

The basic DTW method allows a lot of flexibility for the warping path through the local distance matrix. For example, unlimited vertical and horizontal paths are permissible. These entail that either a single observation in the query is expanded to match a section of the reference or that a section of the query is compressed to match a single observation of the reference. In many contexts, extreme warpings may be unrealistic. To limit such warping various constraints have been proposed on the slope of the path through the cost-distance matrix. This family of constraints, known as local constraints or slope constraints are expressed as an allowable step pattern showing which predecessor points are allowable paths for a given point in the local distance matrix. Sakoe and Chiba [17] devised step patterns based on the maximum allowable number of horizontal or vertical steps that may be taken before a diagonal step must be taken and classified them according to the ratio of these two numbers, the parameter P. They proposed 4 step patterns for $P = 0, 0.5, 1$ and 2, as shown in Fig. 2. We have found it necessary to extend this scheme to obtain arbitrarily strong constraints of $P = 3, 4, 5, \ldots,$ etc., as shown in Fig. 2. The original four step patterns of [17], as well as our extension of this scheme are described in more detail as follows:

- $P = 0$: Infinitely many steps in either the horizontal or vertical directions are permitted regardless of previous steps, i.e. there is no constraint on the slope (equivalent to the basic DTW method).
- $P = \frac{1}{2}$: up to 2 steps in either the horizontal or vertical directions are permitted provided they are preceded by 1 diagonal step.
- $P = 1$: each horizontal or vertical step must be preceded by at least 1 diagonal step
- $P = 2$: each horizontal or vertical step must be preceded by at least 2 diagonal steps
- $P = p$: each horizontal or vertical step must be preceded by at least p diagonal steps

For the mathematical expressions and how these step patterns are incorporated into the dynamic programming algorithm see the original work [17]. The choice of local constraint is in effect a trade-off between trajectory synchronisation, and time distortion. If the constraint is too strong, then the trajectories may not be adequately synchronised, whilst if the constraint is too weak, the trajectories will be closely synchronised but at the cost of unrealistic warpings in the time dimension (extreme contractions or expansions). A method for evaluating this trade-off is the primary goal of this work.

It is not realistic that a choice of local constraint could be made based on process knowledge. Kassidas et al. [6] use DTW without a local constraint, and other authors appear to take the same approach, in some cases using instead a global constraint to limit extreme warpings (to be discussed in section 2.3). To our knowledge, the effects of different local constraints on DTW alignment of batch process data have not been previously investigated. We will demonstrate the advantages of exploiting the local constraint possibilities in DTW to obtain realistic data alignment.

In order to select the best local constraint, we first consider the two extremes: the most lenient constraint, $P = 0$, and the most restrictive constraint which we call $P = p_{max}$. These define the feasible region for choosing the local constraint. The value of p_{max} depends on the length of the query batch K_i relative to the length of the reference batch K_{Ref}. Consider the $K_i \times K_{Ref}$ local distance matrix $\mathbf{C}$. As P increases, the warping function is constrained to a path closer to the diagonal. If $K_i \neq K_{Ref}$ then $\mathbf{C}$ is non-square and a path from $(1,1)$ may consist of at most $\min(K_i, K_{Ref})$ diagonal steps before reaching the edge of the matrix (the first step "onto" (1,1) is conventionally considered a diagonal step). For the path to reach the corner at (K_i, K_{Ref}), requires $|K_i - K_{Ref}|$ additional horizontal or vertical steps. This is illustrated in Fig. 3. In the plot of the 5×8 matrix, there may be at most 5 diagonal steps requiring 3 horizontal steps to form a path from $(1,1)$ to $(5,8)$. This is the case regardless of how the 5 diagonal and 3 vertical steps are distributed. The ratio of the maximum number of diagonal steps possible to the associated number of

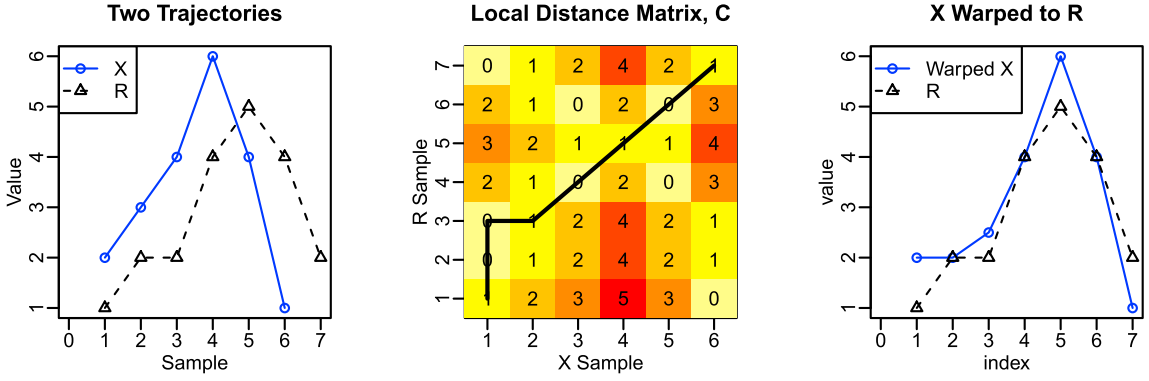

Fig. 1. Two simple trajectories (left), the local distance matrix and the optimum warping path (black line) found by DTW (center), and the resulting warped trajectory X.

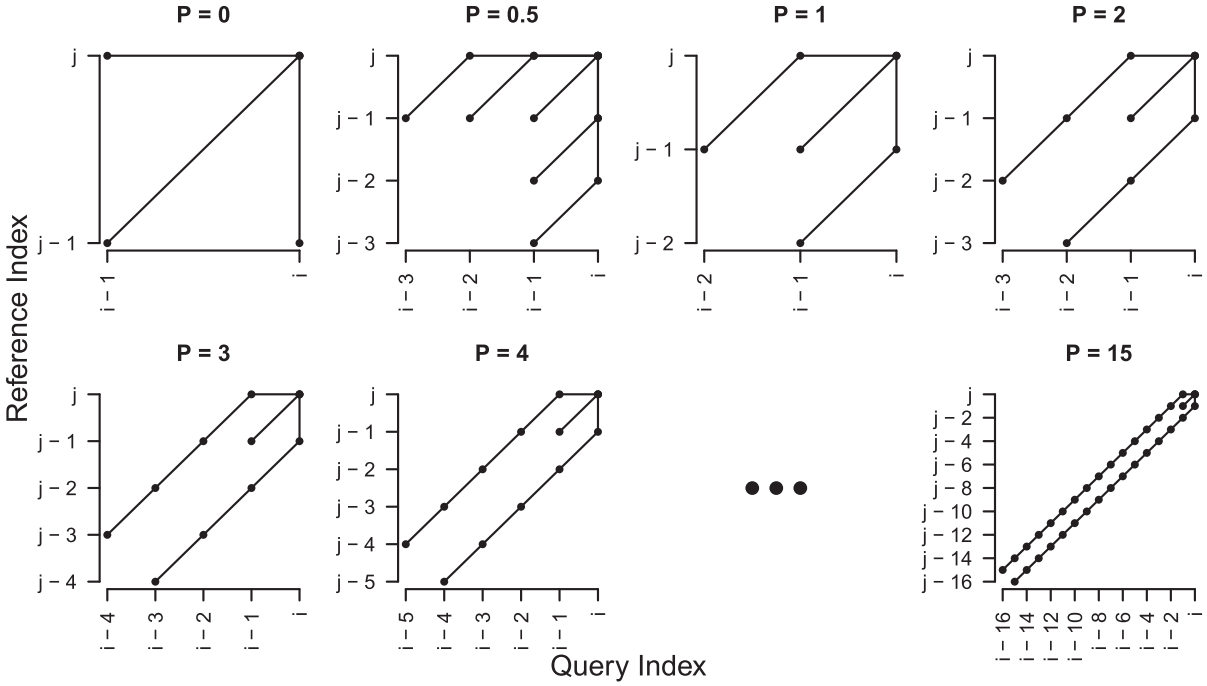

Fig. 2. The four symmetric step patterns proposed by [17] (top row) and our extension of this scheme for arbitrarily large integer P (bottom).

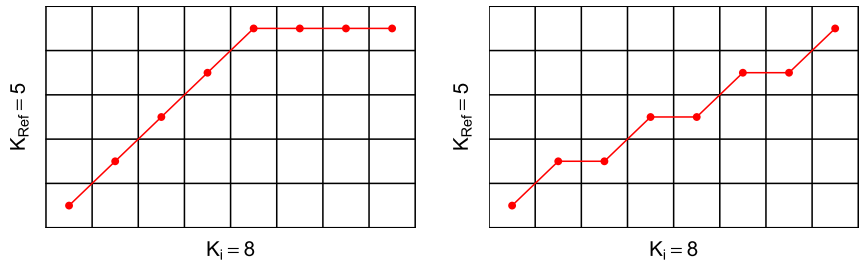

Fig. 3. Two possible warping paths for $K_i = 8$ and $K_{ref} = 5$, showing that the greatest number of diagonal steps possible is 5 and this requires 3 horizontal steps to complete a valid path.

horizontal or vertical steps required provides the greatest P value which still allows a possible warping path to be found. As we cannot allow fractions of a diagonal step, the floor of this ratio is taken to obtain the strictest local constraint for a particular query batch. Once this is done for every query batch in the dataset, p_{max} is taken to be the smallest of the values, so that the same $P = p_{max}$ local constraint can be used on all the batches.

$$p_{max} = \min_{1<i<I} \left\lfloor \frac{\min(K_i, K_{ref})}{|K_i - K_{ref}|} \right\rfloor \tag{9}$$

The $P = p_{max}$ local constraint is the strictest constraint under which every query batch in the dataset can be aligned to the reference batch using DTW, and corresponds to the DTW alignment that is most comparable to linear interpolation of each query batch at K_{ref} sampling points, i.e., minimal warping takes place so events within the batch are not well synchronised. Conversely, the $P = 0$ constraint allows unlimited warping, and the batches are aligned as closely as possible. The problem is to select a local constraint between these extremes that can produce the closest alignment without extreme warpings. To evaluate this trade-off, the following measures are first defined to quantify the two effects.

With regards to degree of variable synchronisation, recall that the DTW distance, Eq. (5), is precisely a measure of the similarity between the query trajectories and the reference trajectories, after DTW synchronisation by the warping function f. In addition, this quantity is returned by the DTW algorithm and so requires no additional computation. To quantify the degree of synchronisation for a particular local constraint $P = p$, align each batch under this constraint. Let D_i be the DTW distance calculated for the i^{th} batch. Define the synchronisation value, $\overline{D}$, as the mean of the DTW distances across the I batches

$$\overline{D} = \sum_{i=1}^{I} D_i / I \tag{10}$$

A small synchronisation value indicates a close alignment of the batches.

To quantify the degree of time distortion, we note that warping only occurs at horizontal and vertical transitions in the warping path whilst diagonal transitions correspond to no change. Therefore, we define the time distortion measure $\overline{N}_{HV}$ as the mean number of horizontal or vertical steps per batch alignment. Denote the warping function for the i^{th} batch by $f^{(i)}(t) = (f_X^{(i)}(t), f_R^{(i)}(t))$, $t = 1, \ldots, T^{(i)}$. A horizontal or vertical step entails that either $f_X^{(i)}(t)$ remains the same for a step, or that $f_R^{(i)}(t)$ remains the same, and the distortion measure is defined as

$$\overline{N}_{HV} = (1/I) \sum_{i=1}^{I} \sum_{t=1}^{T^{(i)}-1} \mathbb{1}_0\left(\left(f_X^{(i)}(t+1) - f_X^{(i)}(t)\right) \cdot \left(f_R^{(i)}(t+1) - f_R^{(i)}(t)\right) \right) \tag{11}$$

where $1_0(x) = \mathbb{1}$ for $x = 0$ and 0 otherwise.

In order to combine $\overline{D}$ and $\overline{N}_{HV}$ into a single "alignment score" for

each local constraint they should be scaled to make them comparable (the range spanned by $\overline{N}_{HV}$ for different local constraints will differ in magnitude to the range spanned by $\overline{D}$). This is done by subtracting the minimum and dividing by the range so each measure varies from 0 to 1. The combined alignment score may then be calculated as the euclidean distance of $(\overline{N}_{HV(\text{scaled})}, \overline{D}_{(\text{scaled})})$ to the origin $(0, 0)$:

$$\text{Alignment Score} = \sqrt{\overline{N}^2_{HV(\text{scaled})} + \overline{D}^2_{(\text{scaled})}} \tag{12}$$

The local constraint which has the smallest alignment score is chosen as the most suitable and the aligned data under this constraint may be used for further analysis. This constraint will synchronise the key events in the process without unrealistic time distortions.

With regards to the variable weights **W**, the above procedure requires that the same variable weights are used for each alignment so that $\overline{D}$ is comparable for different local constraints. We suggest that **W** is calculated prior to local constraint selection, using the method in [16] with the basic DTW algorithm ($P = 0$ local constraint). These variable weights are then used thereafter. In this way, the variable weights are based on the potential warping information, under the most flexible local constraint.

Our proposed method for selecting the local constraint is summarised as follows:

- Determine variable weights **W** by applying the method from [16] with $P = 0$ local constraint.
- Select the reference batch (see section 2.4)
- Calculate the value of p_{max} (Eq. (9))
- For p in $0, \frac{1}{2}, 1, 2, \ldots, p_{max}$, align the batches to the reference batch using DTW with $P = p$ local constraint
- For each different local constraint alignment calculate the synchronisation value $\overline{D}$ and the distortion measure $\overline{N}_{HV}$
- For each local constraint calculate the alignment score by adding the scaled synchronisation value and scaled distortion measure.
- Select that local constraint which results in the smallest synchronisation score for the final alignment of the data

2.3. Global constraint

A global constraint may be specified such that the warping function may not enter certain regions of the local distance matrix. A simple global constraint is the band constraint [17], where the time difference between the warped query and reference series may not exceed some value b: $|f_X(t) - f_R(t)| \leq b$. This means that the warping function is confined to a band of width b along the main diagonal of the distance matrix. The band constraint is problematic when the query and reference are of different lengths, as the main diagonal of a non square matrix does not go from corner to corner.

Alternatively, arbitrary regions of the distance matrix may be selected as off-limits to the warping function. González-Martínez et al. [5] define global constraints for use in their on-line DTW implementation as an empirical envelope derived from historical warping functions in an off-line analysis. Lu et al. [8] use global constraints to avoid warping regions in the data that contain important features. This procedure requires identifying features and selecting which features should not be warped, requiring much analysis and interpretation. It does not seem physically realistic that local batch time would follow the reference time exactly in certain predefined regions, but warp freely in other regions.

Global constraints entail that there is a maximum possible time difference between the warped series. As long as this maximum is not exceeded, extreme compressions and expansions may still occur. This abrupt cut-off in maximum allowable time difference does not seem realistic if the warping function is interpreted physically as the local batch time. It seems unlikely that local batch time could naturally adhere to such a cut-off. In contrast, the local constraint may be physically interpreted as a limit to the rate of change of local batch time relative to the reference, which is more realistic.

Finally, it has been shown that the local constraints for $P > 0$ implicitly limit the region of possible warping paths to the so-called Itakura parallelogram [9], the sides of which have slopes $1/S$ and S where $S = (p+1)/p$ for local constraint $P = p$. For example, local constraint $P = 1$ limits the warping path to the Itakura parallelogram with sides of slope $1/2$ and 2 (Fig. 4). Therefore, we do not see an advantage to introducing additional global constraints if a local constraint is used.

2.4. Other DTW considerations

Our main interest in this work is the use of local and global constraints to control the performance of DTW as outlined in sections 2.2 and 2.3. In this section we consider several issues which are relevant for applying DTW to alignment of batch process data but which are not related to the local or global constraints.

During DTW, each query batch is aligned to the same reference batch in order to obtain aligned batches of the same length as the reference. The choice of reference batch is of importance as the data will be re-expressed in terms of its relation to the reference batch. To aid interpretation of the DTW results, it is preferable that the reference batch be a typical batch found under normal operating conditions. Following [5,6,16], we select the batch with a duration closest to the median duration of the dataset as the reference. It should also be noted that it is important to carry out additional checks that the chosen batch represents normal operating conditions.

Prior to DTW alignment, the data should be scaled so that the local distance measure in Eq. (1) is not influenced by the different engineering units of the variables. Following [6] this is done by dividing each variable by its average range across batches. Ramaker et al. [16] hold that scaling is not necessary for spectroscopic data where each variable has the same unit and similar range. Mean standard deviation or interquartile range could also be used as a scaling factor.

After the data has been scaled, the appropriate weight of each variable, **W** in Eq. (1) must be determined. Variables with a large relative weight will contribute more to the distance and will thereby have greater influence on the warping function and the final alignment. Therefore appropriate weighting will lead to better synchronisation. There have been two proposed methods for defining variable weights. Both are iterative methods. The first method, from Kassidas et al. [6], aims to give large weights to variables that are consistent from batch to batch. These are the variables which can be most closely synchronised by DTW. However, as noted by [16], the method assigns large weights to flat, featureless variables which are actually poor indicators for alignment. For this reason, we choose to use the second method, proposed by Ramaker et al. [16], which assigns large weights to variables which produce a steep valley in the surface plot of the local distance matrix. In section 3.3 we apply both the method from [6] and from [16] and find that the latter results in more intuitive weights based on visual inspection of the data.

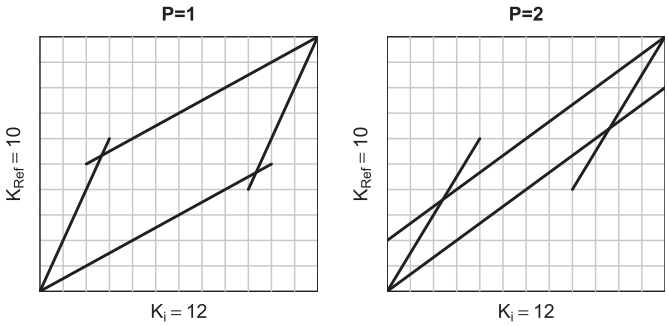

Fig. 4. Itakura parallelograms for the $P = 1$ local constraint, having sides that slope $1/2$ and 2, and the $P = 2$ local constraint, having slides that slope $2/3$ and $3/2$

Finally, we consider the case where batches are stopped at varying degrees of completeness. This may occur in processes where there is no reliable way to measure the completeness of a batch exactly. In this case, some batches will contain process events which are not present in other batches. One approach to aligning such data is to use partial DTW alignment in which the endpoint constraints of Eq. (6) are relaxed to allow for all of $\mathbf{X}$ to be aligned to a section of $\mathbf{R}$. The DTW solution may be interpreted as performing K_{ref} DTW alignments, using $\mathbf{R}(1:k,:)$ as the reference for $k = 1, \ldots, K_{ref}$. Then the alignment which gives the minimum value of $D_f(\mathbf{X}, \mathbf{R}(1:k,:))$ is adopted and indicates that the batch completion of $\mathbf{X}$ corresponds to the first k observations of $\mathbf{R}$. Similarly, a partial alignment can account for uncertainty concerning the start time of the batches, or of both start time and end time. Of course, it is necessary that a reference batch which is "more complete" than all the query batches is chosen.

A disadvantage of conducting a partial alignment is that after synchronisation, the batches will still have varying lengths. Secondly, in omitting the start point or end point constraint, the algorithm is given a great deal more flexibility which may result in inappropriate synchronisation. The constraint of fixed endpoints is a great advantage, (if the start and end endpoints do indeed represent the same events in all batches) as they ensure that at least these points are appropriately aligned, leading to a greater confidence in the alignment of the intervening points. With an open ended alignment, there is uncertainty whether the reference section selected by DTW truly represents the events in the query, or whether a chance match was found to a section containing different events, because of atypical behaviour in the batch.

We suggest dealing with varying completion amounts as follows:

1. Using process knowledge, identify a feature in one or more of the variable trajectories, near the end of the batch, which is present in all batches and which may be assumed to represent the same event across batches. Truncate each batch at the time of this end-point event.
2. Select local constraint using global DTW and synchronise data using the optimum local constraint.
3. For subsequent analysis, each synchronised batch matrix may be appended with the very last observation from the unsynchronised batch matrix in order to represent the information on what took place after the endpoint event for each batch. As a single observation is used for this, all the synchronised batches will still have the same length.

A similar method could be used if the start-point of the batches varies. If an end or start event cannot be identified, then partial DTW may be of value.

3. Results and discussion

The methods described in section 2 were applied to real data using the statistical software R [15]. To implement DTW with various local constraints, global constraints, variable weights etc. the DTW R package was used [4]. The Sakoe and Chiba [17] local constraints for $P = 0, 0.5, 1$ and 2 are built in to this package, and the additional constraints described in section 2.2 for integer $P > 2$ may be readily defined by the user.

3.1. The data

Data was provided by the company Chr. Hansen for an industrial bacteria fermentation process used to produce a multi-strain bacteria culture for use in the dairy industry. Process data was obtained for $I = 23$ batches consisting of $J = 6$ variables measured throughout the batch duration. The raw data was processed to obtain measurements at evenly spaced time intervals resulting in $403 \leq K_i \leq 532$ observations. This processed data is shown in Fig. 5. A batch begins with the introduction of bacteria cells to a fermentation vessel which has been pre-filled with growth media. As the bacteria cells grow and multiply they produce acid which lowers the pH inside the fermenter. Once the pH reaches a predefined set point roughly half-way through the process, a controller is

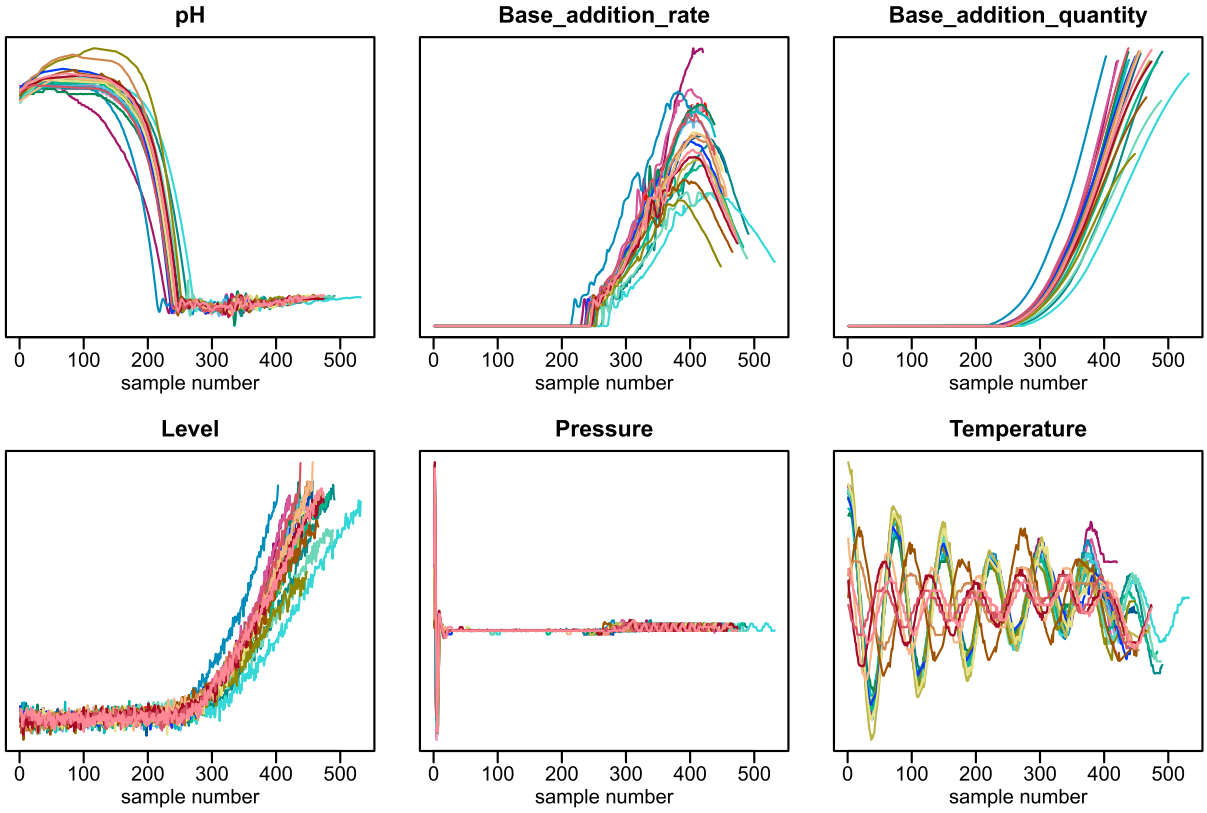

Fig. 5. The 23 batches in the dataset obtained from the batch fermentation process. The scale of the ordinate axis is omitted for confidentiality reasons.

activated which is designed to maintain the pH at this set point for the remainder of the batch. The controller does this by manipulating the amount of base (ammonia) added to the fermenter in response to changes in measured pH. The rate of base addition is measured, as well as the total quantity of base added to the fermenter. The level inside the tank rises as base is added. Temperature is kept within a narrow range. The point at which to stop the batch is decided by the operator based on process knowledge (judgement on the degree of bacteria growth based on amount or current rate of base addition) and logistics (e.g. availability of equipment downstream). In this process, the pH and base addition rate/quantity can be assumed to be direct indicators of the biological state of the process. Level is closely correlated to base addition. Pressure is kept constant. However, it was determined that the temperature variable is not a reflection of the progress of bacteria growth, but rather depends on the physics of maintaining the temperature of a large body of fluid, and the mixing of this fluid in relation to the sensor placement. Therefore, the temperature variable is not a direct indicator of the biological progress of the bacteria fermentation. Based on this prior knowledge, the temperature variable should not influence the alignment of the data.

Time variation between batches is clearly visible in the process. As well as the overall batch duration which varies between 403 and 523 observations, there is also variation in the timing of the events that take place during the process. The lag phase (the time before the bacteria multiply enough to decrease the pH) varies from around 50, to around 200 observations. The time at which the pH reaches the set-point varies from 200 to 270 observations. The time at which base addition rate peaks varies between 350 and 450. In addition to these more visible events in the biological process, we may suppose that there are many other features in the variable trajectories that reflect other events in the process, and these also vary in time from batch to batch.

3.2. Processing data and selecting reference batch

The batches have been stopped at varying degrees of completion, as illustrated by how complete a curve is traced by the base addition rate trajectory after it has peaked. The last observation in each batch does not correspond to the same event, and so should not be synchronised to each other. The methods proposed in this paper determine the most appropriate constraints to use for DTW, by applying the global alignment version of DTW. Therefore, in cases like this, some endpoint that is common to all batches must be inferred in order to truncate the data so that all batches stop at the same event. The time of reaching peak base addition rate was selected as the imposed endpoint. This time was determined by fitting a cubic polynomial to the last section of the base addition rate trajectory, and the batches were truncated at this point.

Next, the reference batch was selected to be that batch with duration closest to the median duration, having confirmed with operators that the selected batch had desirable properties. The data was scaled by dividing each variable by its mean range across all batches and centred by subtracting the minimum (across all batches).

The truncated data and reference batch are shown in Fig. 6.

3.3. Determining variable weights

The variable weights are determined by applying the method from [16] with unconstrained DTW ($P = 0$) and initial weights of 1 for all variables (Temperature excluded as previously discussed). After seven iterations the weights converged ($\varepsilon < 0.002$) to the values shown in Fig. 7. It is reasonable that pH is given the greatest weight as it is the main indicator of the state of the process. Base addition quantity and rate are also important indicators, but are only relevant for the second half of the process so receive less weight. Pressure and Level contain very little information on the state of the process and so it is appropriate that they are given small weights.

The weights that result from the method in [6] are also shown in Fig. 7. This method required 19 iterations before $\varepsilon < 0.002$ and it weighted pressure and level greater than base addition rate and base addition quantity. This contradicts our prior interpretation that level and

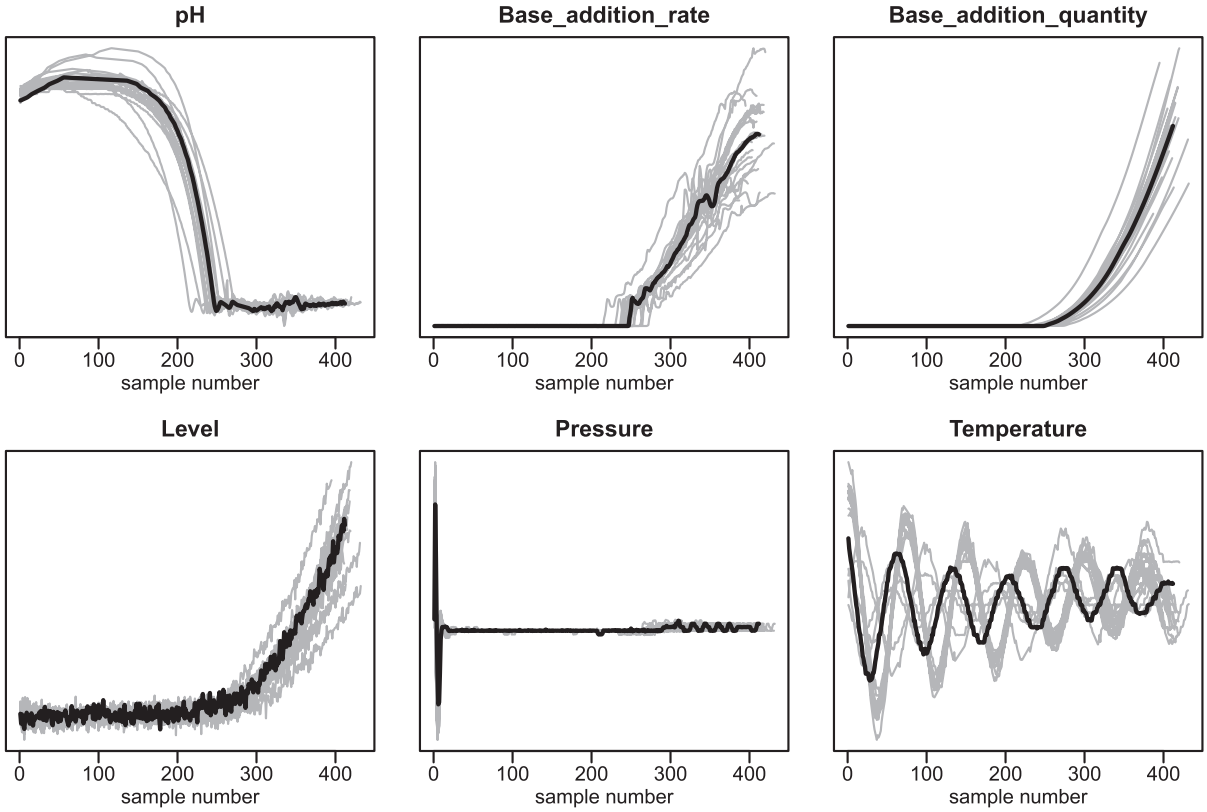

Fig. 6. The truncated data with reference batch in black and other batches in grey.

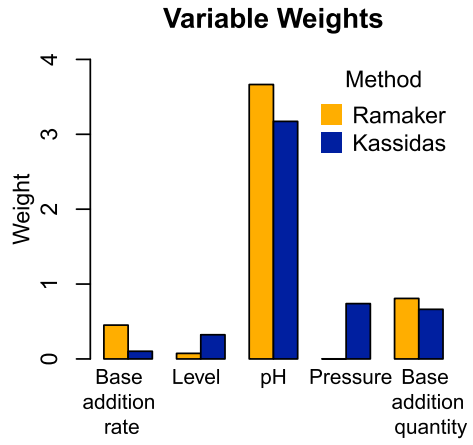

Fig. 7. Variable weights resulting from the methods of Ramaker et al. [16] and Kassidas et al. [6].

pressure are the least informative variables for alignment and supports our choice of the Ramaker method for determining variable weights.

3.4. Selecting local constraint

After truncating the batches to a common end point, they range in length from 387 to 432 observations. The reference batch is of length 412 observations. From Eq. (9), the strictest constraint which can be applied to all batches is $P_{max} = 15$. Therefore a choice must be made amongst the constraints $P \in [0,\ 0.5,\ 1,\ \ldots,\ 15]$. DTW alignments were applied for each of these 17 local constraint choices. The resulting time warping functions and aligned trajectories of the pH variable are shown for $P = 0, 3$ and 15 in Fig. 8.

Fig. 8 clearly shows that in the traditional DTW method where $P = 0$ and there is no constraint on the slope of the warping function, there are many extreme warpings in the time warping functions: long sections of the query batch are compressed to an instant or a single observation in the query batch may be expanded to cover long sections of the reference batch. The variable trajectories are aligned too aggressively by traditional DTW. The strongest constraint, $P = 15$ approaches similar results to a linear interpolation. The aligned batches have the same number of observations as the reference, but events during each batch are not synchronised. The intermediate constraint shown in Fig. 8 ($P = 3$), results in much less extreme warping than for $P = 0$, whilst appearing to succeed in synchronising the key events in the pH variable.

For each local constraint alignment, $\overline{N}_{HV}$ (degree of time distortion) and $\overline{D}$ (synchronisation value) was calculated. The statistics are scaled and combined to form the alignment score. The constraint resulting in the smallest alignment score was the $P = 3$ constraint, although $P = 2$ has a score very close to it. This procedure is shown in Fig. 9.

With no local constraint ($P = 0$, i.e. traditional DTW), the closest variable synchronisation is obtained, but also the greatest time distortion, $\overline{N}_{HV} \approx 500$. As the local constraint is strengthened up to $P = 3$, time distortion is reduced significantly to $N_{HV} < 100$, without much deterioration in variable synchronisation (small increase in $\overline{D}$). Further strengthening of the local constraint up to $P = 15$ does not lead to substantial reduction in time distortion but results in rapid deterioration in variable synchronisation. This relationship results in an "elbow" in the plotted values which could form the basis for choosing P without the need for scaling and score calculation steps. The "elbow" indicates the choice of P at which strengthening the constraint further will result in substantially poorer alignment with little reduction in time distortion. This heuristic may be summarised as follows:

- Plot $\overline{D}$ against $\overline{N}_{HV}$
- Locate the bend in the plot and select the smallest local constraint closest to the bend

This approach would lead us to select $P = 3$ or $P = 2$. Although the

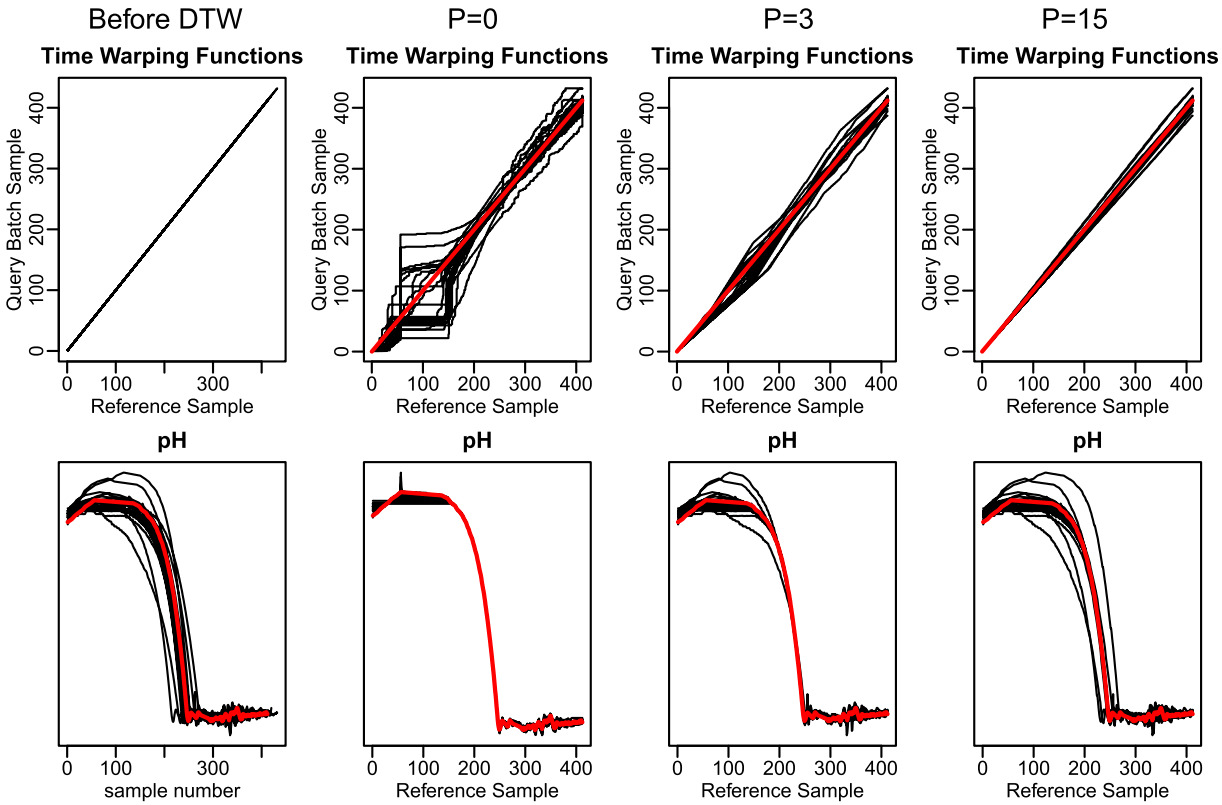

Fig. 8. The warping functions (top) and the aligned trajectories for the pH variable (bottom) are shown from DTW alignment with local constraints of $P = 0, 3$ and 15 (left to right).

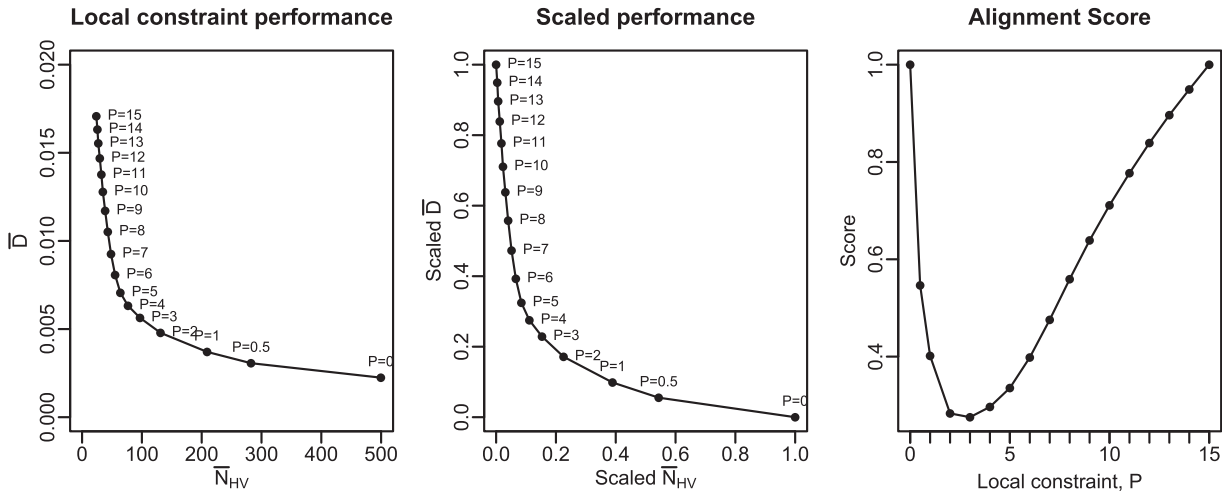

Fig. 9. Choosing the local constraint parameter, P, based on the alignment performance according to $\overline{N}_{HV}$ and $\overline{D}$ (left), their scaled counterparts (center) and the alignment score (right).

method is more subjective, in this instance it leads to the same conclusion as the score method.

In summary, the above off-line analysis has determined the variable weights and the appropriate local constraint for DTW alignment of this data. Using these parameters, the data may then be aligned for further analysis by multi-way methods. In addition, these parameter settings can be used on new batch data from the process, or even in applying DTW in an on-line application.

3.5. Comparison to DTW with global constraint

For comparison, the effect of using a global constraint was investigated by applying the "slanted band" global constraint from [4]. This global constraint limits the warping function to a band that is b columns wide and is centred on the line joining start-corner and end-corner of the local distance matrix. The data was aligned with the slanted band global constraint for bandwidths of $b = 5, 10, 20, 40$ and 80 without local constraint ($P = 0$ for all b). For each bandwidth, time distortion value $\overline{N}_{HV}$, as well as variable synchronisation value $\overline{D}$ was calculated. The resulting values are compared to the previous results from varying the local constraint in Fig. 10. The global constraint is not as effective as the local constraint in reducing time distortion, and strengthening the global constraint (decreasing b) leads to rapid deterioration in variable synchronisation (increase in $\overline{D}$) without much reduction in time distortion. We conclude that the local constraint approach is superior to the slanted band global constraint at limiting unrealistic warping without serious loss in variable synchronisation quality.

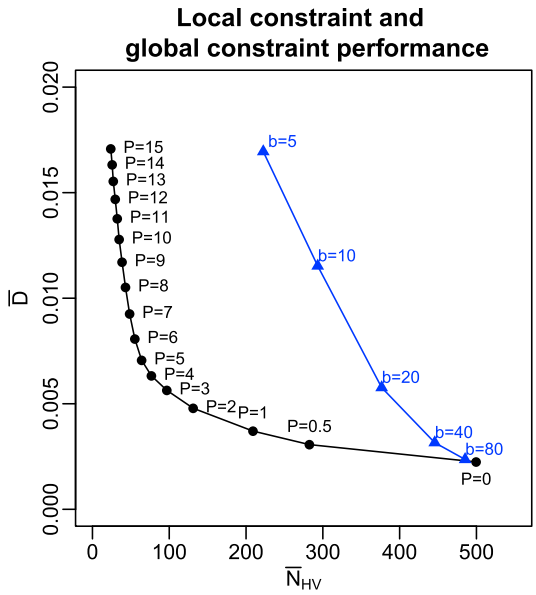

Fig. 10. $\overline{N}_{HV}$ and $\overline{D}$ are shown for varying local constraint (black dots) and varying width of the slanted band global constraint (blue triangles). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

4. Conclusions

DTW has been widely used for the alignment of batch process data, though there is not a single accepted approach to how the algorithm should be applied. We have investigated the use of local constraints, an area which has received little attention previously in this application. We have shown how the local constraint may be used to avoid unrealistic warping in the aligned data and presented a novel method for selecting the most appropriate local constraint for an alignment problem. This method optimises the trade-off between variable synchronisation and time distortion. We maintain that the problem of unrealistic warping in DTW should be addressed using the proposed local constraint method, rather than a global constraint approach for two main reasons. Firstly, the global constraint does not avoid extreme contractions or expansions less than some upper limit, whilst the local constraint method limits extreme warping more pervasively. Secondly, the local constraint method we propose is very generalizable to other datasets, whilst many global constraint methods require much tailoring to the dataset at hand. By adopting the proposed method to select an appropriate local constraint, batch process data may be aligned realistically, allowing the warping functions of the aligned batches to be interpreted as progress signatures of the batches.

Acknowledgements

The project received financial support from Innovation Fund Denmark through the BIOPRO2 strategic research center (Grant number 4105-00020B). We thank Chr. Hansen A/S for providing access to data and expertise on the production process.

References

[1] M. Fransson, S. Folestad, Real-time alignment of batch process data using cow for on-line process monitoring, Chemom. Intell. Lab. Syst. 84 (1) (2006) 56–61.

[2] S. García-Muñoz, T. Kourti, J.F. MacGregor, A.G. Mateos, G. Murphy, Troubleshooting of an industrial batch process using multivariate methods, Ind. Eng. Chem. Res. 42 (15) (2003) 3592–3601.

[3] S. García-Muñoz, M. Polizzi, A. Prpich, C. Strain, A. Lalonde, V. Negron, Experiences in batch trajectory alignment for pharmaceutical process improvement through multivariate latent variable modelling, J. Process Control 21 (10) (2011) 1370–1377.
[4] T. Giorgino, et al., Computing and visualizing dynamic time warping alignments in r: the dtw package, J. Stat. Softw. 31 (7) (2009) 1–24.
[5] J.M. González-Martínez, A. Ferrer, J.A. Westerhuis, Real-time synchronization of batch trajectories for on-line multivariate statistical process control using dynamic time warping, Chemom. Intell. Lab. Syst. 105 (2) (2011) 195–206.
[6] A. Kassidas, J.F. MacGregor, P.A. Taylor, Synchronization of batch trajectories using dynamic time warping, AIChE J. 44 (4) (1998) 864–875.
[7] D. Louwerse, A. Smilde, Multivariate statistical process control of batch processes based on three-way models, Chem. Eng. Sci. 55 (7) (2000) 1225–1235.
[8] B. Lu, S. Xu, J. Stuber, T.F. Edgar, Constrained selective dynamic time warping of trajectories in three dimensional batch data, Chemom. Intell. Lab. Syst. 159 (2016) 138–150.
[9] M. Müller, Information Retrieval for Music and Motion, vol. 2, Springer, 2007.
[10] C. Myers, L. Rabiner, A. Rosenberg, Performance tradeoffs in dynamic time warping algorithms for isolated word recognition, IEEE Trans. Acoust. Speech, Signal Process. 28 (6) (1980) 623–635.
[11] D. Neogi, C.E. Schlags, Multivariate statistical analysis of an emulsion batch process, Ind. Eng. Chem. Res. 37 (10) (1998) 3971–3979.
[12] N.-P.V. Nielsen, J.M. Carstensen, J. Smedsgaard, Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping, J. Chromatogr. A 805 (1) (1998) 17–35.
[13] P. Nomikos, J.F. MacGregor, Monitoring batch processes using multiway principal component analysis, AIChE J. 40 (8) (1994) 1361–1375.
[14] P. Nomikos, J.F. MacGregor, Multi-way partial least squares in monitoring batch processes, Chemom. Intell. Lab. Syst. 30 (1) (1995) 97–108.
[15] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2014. http://www.R-project.org/.
[16] H.-J. Ramaker, E.N. van Sprang, J.A. Westerhuis, A.K. Smilde, Dynamic time warping of spectroscopic batch data, Anal. Chim. Acta 498 (1) (2003) 133–153.
[17] H. Sakoe, S. Chiba, Dynamic programming algorithm optimization for spoken word recognition, Acoust. Speech Signal Process. IEEE Trans. 26 (1) (1978) 43–49.
[18] G. Tomasi, F. Van Den Berg, C. Andersson, Correlation optimized warping and dynamic time warping as preprocessing methods for chromatographic data, J. Chemom. 18 (5) (2004) 231–241.

CHAPTER 4

Predicting batch harvest time with Chr. Hansen

The collaboration with Chr. Hansen continued with a second topic of research. Prediction of product quality was of great interest. A number of product quality parameters were provided, and a preliminary analysis performed to predict these measurements based on the process data with, for example, multiway PLS. However, results were not encouraging. It was found that the range spanned by these quality parameters was small, and the models were therefore unable to explain the small differences in quality between different batches. Ideally, a model would be built using a dataset of batches ranging from poor quality to excellent quality. However, obtaining such a dataset from production data is not realistic, as in actual production great pains are taken to ensure that the quality parameters for every batch falls within a narrow range defined by product specifications. This challenge was encountered once more during the collaboration with CP Kelco discussed in Chapter 5.

As prediction of the product quality parameters related to product specifications was not feasible, the direction of research once more turned toward the time variation in the process. Variation in the duration of batches was observed, due both to variation in the rate of growth of the bacteria as well as variation in when the process operator decided to stop the batch. A need for predicting batch harvest time (the time at which a batch is stopped) was identified. If harvest time could be predicted at an early stage in an ongoing batch, then the ability to schedule future batches and allocate down-stream resources would be greatly improved. A method was developed that used the time information provided by dynamic time warping, to forecast the future harvest time of an ongoing batch. This method was novel in making literal use of the dynamic time warping function for predicting how quickly or slowly a batch will be completed. In addition, lasso regression was incorporated in order to select

the relevant process data for predicting harvest time. The journal article based on this research is included in the following section.

4.1 Paper 2: Harvest time prediction for batch processes

Spooner, M., D. Kold, and M. Kulahci. 2018. "Harvest time prediction for batch processes". *Computers & Chemical Engineering* 117:32-41.
https://doi.org/10.1016/j.compchemeng.2018.05.019.

Harvest time prediction for batch processes

Max Spooner[a,*], David Kold[b], Murat Kulahci[a,c]

May 22, 2018

[a]DTU Compute, Technical University of Denmark, Kgs. Lyngby, Denmark
[b]Chr. Hansen A/S, Hvidovre, Denmark
[c]Department of Business Administration, Technology and Social Sciences, Luleå University of Technology, Luleå, Sweden
*Corresponding Author. Address: DTU Compute, Asmussens Alle 322, 2800 Kgs. Lyngby, Denmark. E-mail: mpsp@dtu.dk (M. Spooner)

Keywords: Batch Process, Prediction, Dynamic Time Warping, Partial Least Squares, Lasso Regression

Abstract

Batch processes usually exhibit variation in the time at which individual batches are stopped (referred to as the harvest time). Harvest time is based on the occurrence of some criterion and there may be great uncertainty as to when this criterion will be satisfied. This uncertainty increases the difficulty of scheduling downstream operations and results in fewer completed batches per day. A real case study is presented of a bacteria fermentation process. We consider the problem of predicting the harvest time of a batch in advance to reduce variation and improving batch quality. Lasso regression is used to obtain an interpretable model for predicting the harvest time at an early stage in the batch. A novel method for updating the harvest time predictions as a batch progresses is presented, based on information obtained from online alignment using dynamic time warping.

1 Introduction

Batch processes are characterised by a beginning, when the raw materials are loaded into a reactor vessel, a finite period of transformation or growth, and an end when the finished product is harvested from the reactor. The time at which to harvest the batch is often defined based on some features in the process which from experience ensure the desired product specifications. There is often batch to batch variation in the time at which the harvest criterion occurs, and so batches have different durations. This is especially the case in bio-based industrial processes where the harvest time is dependent on the activity of living organisms. In order to ensure the batch is harvested at the optimum point in time, it must be monitored closely by a technician who must react quickly when the harvest criterion is reached. In this work we consider the problem of predicting the harvest time at an early stage in the process. Obtaining good harvest time predictions is of value for two reasons. Firstly, such predictions provide a guide for the technicians on when to focus on the process and when it is safe to work on other tasks. Secondly, the predictions facilitate scheduling of downstream processes such as packaging.

The general problem of predicting some response variable based on data measured during a batch process has been investigated extensively. The predominant approach, Multi-way Partial Least Squares (MPLS), was pioneered by [1] where the method was used to make predictions of 5 quality variables using batch process data. This approach has been applied and adapted by several authors including [2], [3] and [4]. Besides MPLS, a wide range of machine learning methods have been applied to the problem of predicting end-of-batch quality using online process data including neural networks [5], support vector regression [6] and lasso regression [7].

There is limited research on batch process prediction where the response variable is harvest time of the batch rather than end-of-batch quality. In [8] MPLS is used to predict the end time of a batch based on only the first two hours of process data. Others apply more ad-hoc methods to detect the optimal fermentation time of a batch process [9].

In this paper we present a real case study of predicting the harvest time of a batch bacteria fermentation process for the bioscience company Chr. Hansen A/S using a novel statistical approach, and apply the methods to datasets from two different bacteria fermentations. A need for more accurate harvest time prediction was identified by the company. The process is characterised by two phases which influenced our approach for harvest time

prediction. In brief, the proposed method consists of predicting the harvest time at the end of the first phase using a lasso regression model, and then updating the predictions from this model online during the second process phase using the time-information provided by online dynamic time warping (DTW). An overview of the steps involved in the method is shown in Fig. 1. Each step is explained in detail in Section 2

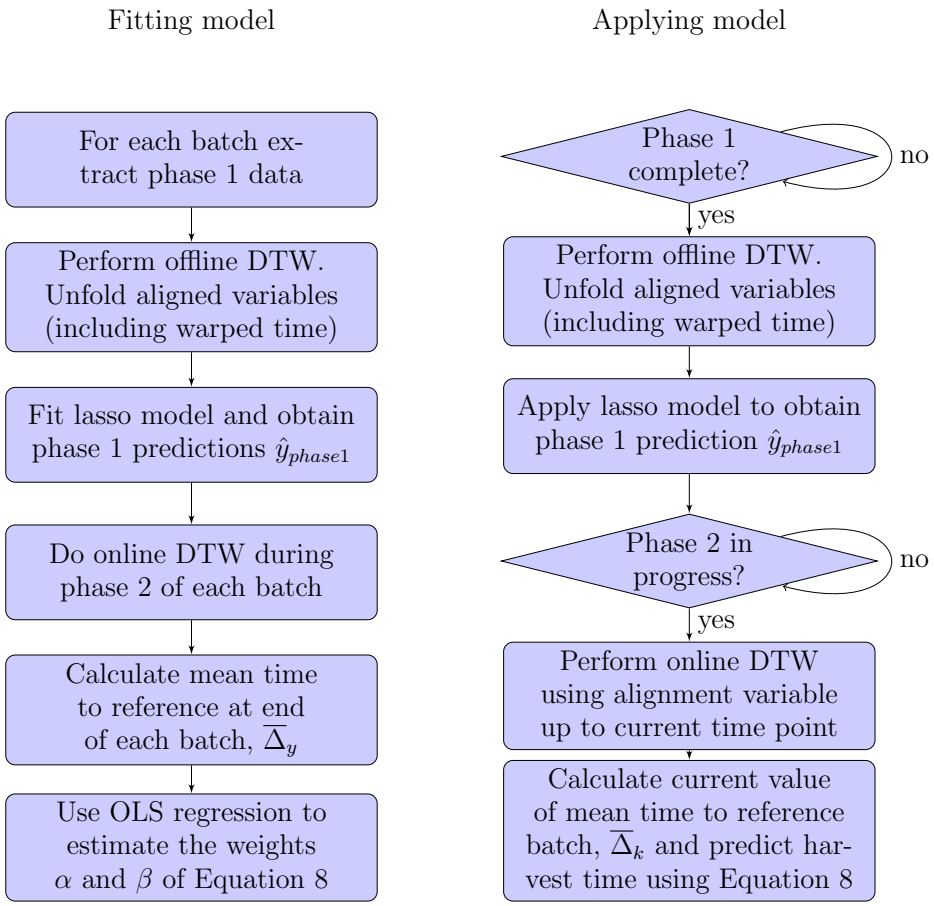

Figure 1: Summary of model fitting and application

The proposed method provides a valuable early warning of the expected harvest time to enable scheduling of downstream tasks. As a benchmark, we compare the phase 1 lasso model to a MPLS model, which is the more usual approach to dealing with the high dimensionality of the unfolded phase 1 data. We advocate the use of lasso regression for two main reasons. Firstly,

lasso regression results in a far simpler model than MPLS, as it automatically selects only a subset of variables for which to give non-zero model coefficients. The lasso model is therefore easier to interpret than the PLS model, which is a big advantage for implementation in an industrial setting. Secondly, the test error, estimated using nested cross validation [10], is found to be smaller for the lasso model than for the MPLS model. Therefore, in this case study, the lasso model is expected to provide more accurate harvest time predictions than the MPLS model when used on new batches.

In summary, the novel contributions of this work consist firstly, in the presentation of real data from an existing batch process and a real problem to be solved: that of predicting the harvest time. Secondly, we demonstrate the advantages of lasso regression as a variable selection method, in contrast to the more predominant latent structure method MPLS that retains all variables in the model regardless of their relevance. Thirdly, we present a novel method that updates the harvest time predictions using dynamic time warping. This method exploits directly the ability of DTW to indicate time information regarding a batch, whereas previous applications for batch processes has mostly used dynamic time warping as a preprocessing step.

2 Methods

2.1 Process

The fermentation process used by Chr. Hansen to produce bacteria cultures consists of the following steps

1. A small volume of concentrated bacteria cells is added to a fermenter vessel which has been pre-filled with growth medium

2. The bacteria cells grow and multiply, thereby producing acid which lowers the pH inside the fermenter

3. When pH reaches a predefined set point, the pH level is automatically controlled at the set point level by adding a base to the fermenter

4. Based on expert judgement the batch is stopped and the contents of the fermenter are transferred to downstream processing

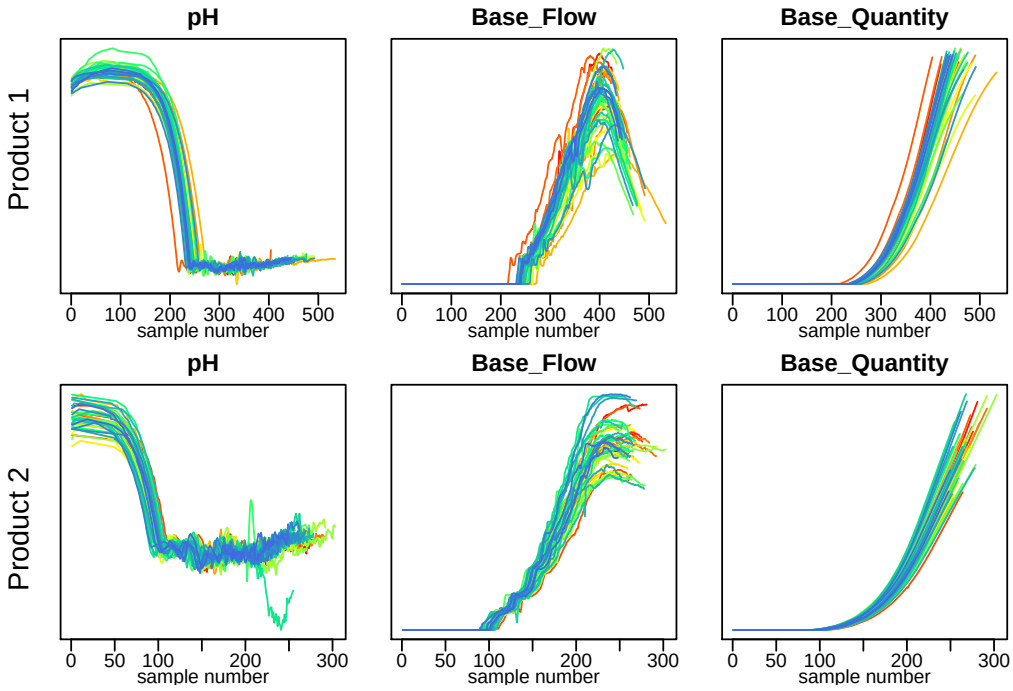

Figure 2: The raw data for both products

Data for two different bacteria fermentations was obtained which we refer to as product 1 and product 2. For both products, 6 variables are measured during the fermentation: pH, Base Flow (rate of base addition), Base Quantity (total amount of base added), Temperature, Level and Pressure. However, our interest was limited to the first three of these variables because they are most closely linked to the biological process. The temperature variable was not used because it was tightly controlled and maintained at a constant level. The product 1 and product 2 data consisted of 44 and 43 batches respectively and were all taken from normal operating conditions. The data is shown in Figure 2 where the two phases of the process can be distinctly seen. In the first phase, pH is not controlled and is decreasing until it reaches the set point. In the second phase, base is added in order to maintain the pH at the set point. The observed changes in pH and Base Flow are closely linked to the state of the process and reflect rates of bacteria growth and metabolism. There is variation between batches in the time taken to reach different stages, as well as in the magnitudes of the variables, due to differences in raw materials and inoculation material.

The criterion to start the harvest is product specific and is based on a combination of process parameters. The harvest process should be initiated manually when the criterion is met, but may be delayed as it relies on human judgement and taking additional factors into account, such as whether equipment downstream is available.

Due to the above mentioned variations and equipment limitations the harvest is done inconsistently as reflected in Figure 2. Therefore, the aim of this work was to predict in advance the time at which the harvest criterion will be met. This will reduce the need for human evaluation thereby enabling less process variation and better planning in regards to utilization of the downstream equipment.

Batches which do not attain the harvest criterion cannot be used for model building, as for such batches the response variable, correct harvest time, is missing. Therefore, due to the batch to batch variation, the harvest criterion for each product was redefined to features attained by all batches so that it would be possible to assess the performance. The harvest criterion for product 1 was defined as the moment when Base Flow falls to 95 % of its maximum value. For product 2 the harvest criterion was defined to be when Base Quantity reaches 0.72 (in scaled units). These criteria are very similar to those used in practice by the company, but have the advantage of being attained by all the batches. Of course, an alternative approach would

be to use the harvest criteria actually in use, and use some missing data imputation method to fill in the correct harvest time values for batches which were harvested too soon. However, then assessment of model performance would depend on how the missing data are imputed. Therefore, we use the former approach so that there is no doubt that the harvest time to be predicted corresponds to the same criterion in all batches.

2.2 Phase 1 Models

The clear division of the process into two phases suggested a goal of predicting the harvest time at the end of the first phase. A prediction at this point is early enough to be useful for scheduling purposes, whilst enough data has been accumulated to make predictions realistic. In [11] approaches to statistical monitoring of multiphase and multistage batch processes are discussed and the idea of monitoring at different levels (phases/stages/entire batches) is presented. We suppose that data is available up to the end of the first phase, and use this data to train a model for predicting the harvest time. The model may then be used during future batches upon completion of the first phase in order to predict when the entire batch will be ready to harvest. Each data set consists of I batches, J variables measured for K_i observations where K_i varies depending on the duration of phase 1 for the i^{th} batch. The response variable is the $I \times 1$ vector of harvest times for the product.

2.2.1 Alignment

The methods we wish to apply require that each batch have the same number of observations. It is also preferable that events during the batch are synchronised [12, 13]. We applied dynamic time warping (DTW) to align the data. This dynamic programming algorithm, originally developed for speech recognition [14], has been used widely for alignment of batch process data [12, 15, 16] as well as in almost every other data analytic field concerned with time series data [17]. With DTW, a reference batch is selected and the other batches are aligned to this reference batch. The aligned batches all have the same number of observations as the reference batch, K_{ref}. In addition to the aligned variable trajectories, a warping function is obtained which represents the local batch time of each batch relative to the reference batch, and this warping function should be treated as an additional variable of the process. As phase 1 is completed when the harvest time predictions are to be made,

we use the offline version of DTW in which end-point constraints enforce the first and last observation of each query batch to be aligned to the first and last observation respectively of the reference batch.

During phase 1, pH is the only variable closely related to the underlying process and is therefore used as the alignment variable. The reference batch is selected as the batch with duration closest to median duration. In order to avoid pathological warpings a local constraint as defined by [14] is used. These local constraints are represented by the parameter P with $P = 0$ corresponding to no constraint on the warping path, and larger values restricting the warping path closer to the diagonal. The value of P can be selected according to the methods in [13].

After alignment, each phase 1 data set is a three way data cube of I batches by J variables (now including the time warping function variable) by K_{ref} observations.

2.2.2 MPLS

Since [1], MPLS has become a standard method for making prediction models from batch process data. In MPLS, the three-way batch process data is first converted to two way data so that the classic PLS method [18] can be applied. This unfolding consists of placing K_{ref} many $I \times J$ time slices side by side to create the matrix $\mathbf{X}$ $(I \times JK_{ref})$. The response variable to be predicted (the final harvest time) forms the vector $\mathbf{y}$ $(I \times 1)$. Each column of $\mathbf{X}$ is mean centred and scaled to unit variance. $\mathbf{y}$ is also mean centred. We opt not to scale $\mathbf{y}$ so that the errors of the model will be in the original units. PLS then de-constructs both $\mathbf{X}$ and $\mathbf{y}$ into scores and loadings in such a way that the covariance between the scores is maximised within the following outer relationship:

$$\mathbf{X} = \mathbf{TP}' + \mathbf{E} \text{ and } \mathbf{y} = \mathbf{UQ}' + \mathbf{F}^* \tag{1}$$

$\mathbf{y}$ is predicted by the scores of $\mathbf{X}$ through the inner relationship

$$\mathbf{y} = \mathbf{TBQ}' + \mathbf{F} \tag{2}$$

where the algorithm minimises $||\mathbf{F}||$. The different elements of the PLS decomposition are summarised in Table 1. The number of components (latent variables) in the PLS model, n, may be selected using cross validation. Problems of collinearity and overfitting may be overcome by selecting a limited number of components, n to retain in the model. Cross validation is

used to select the value of n which results in the smallest cross validation mean squared error, then this value is used to train the model on the entire data set.

Table 1: PLS and Lasso Notation

Symbol	Definition
$\mathbf{X}$	$(I \times JK_{ref})$ Unfolded process data
$\mathbf{y}$	$(I \times 1)$ Response variable
$\mathbf{T}$	$(I \times n)$ PLS scores of $\mathbf{X}$
$\mathbf{U}$	$(I \times n)$ PLS scores of $\mathbf{Y}$
$\mathbf{P}$	$(JK_{ref} \times n)$ PLS loadings of $\mathbf{X}$
$\mathbf{Q}$	$(1 \times n)$ PLS loadings of $\mathbf{y}$
$\mathbf{E}$	$(I \times JK_{ref})$ PLS errors of $\mathbf{X}$
$\mathbf{F}^*$	$(I \times 1)$ PLS errors of $\mathbf{y}$ in outer relation
$\mathbf{F}$	$(I \times 1)$ PLS errors of $\mathbf{y}$ in inner relation
$\mathbf{B}$	$n \times n$ PLS regression coefficients relating scores of $\mathbf{X}$ to scores of $\mathbf{y}$
n	Number of latent variables retained in PLS model
$\boldsymbol{\beta}$	$(JK_{ref} \times 1)$ Lasso coefficients
$\mathbf{e}$	$(I \times 1)$ Lasso errors
λ	Lasso sparsity parameter

2.2.3 Multi-way Lasso

Another approach to dealing with the large numbers of correlated variables in batch process data, is to use a form of regularised regression such as lasso regression. Lasso regression has not been as widely applied to batch process data as PLS, but [19] developed a lasso framework for fault diagnosis, and [7] adapted lasso regression for multiphase batch processes. Lasso regression was developed by [20] and the method finds a linear model

$$\mathbf{y} = \mathbf{X}\boldsymbol{\beta} + \mathbf{e} \tag{3}$$

where the coefficients $\boldsymbol{\beta}$ are calculated based on the minimisation of

$$\min_{\boldsymbol{\beta}} \frac{1}{I}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_1 \tag{4}$$

where $||\cdot||_1$ is the L_1-norm and $||\cdot||_2$ the L_2-norm. λ is a tuning parameter determining the overall influence of the L_1 penalty on the solution to the minimisation. The L_1-norm penalty has the effect of shrinking coefficients in $\boldsymbol{\beta}$ to zero, and the greater the value of λ, the more sparse the coefficient vector will be.

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \frac{1}{I}||\mathbf{y} - \mathbf{X}\boldsymbol{\beta}||_2^2 + \lambda||\boldsymbol{\beta}||_1 \tag{5}$$

Again, as $\mathbf{X}$ is a matrix, the three-way batch process data must be unfolded as described in the previous section. The unfolded matrix is centred and scaled. Cross validation is used to select the value of λ which results in the smallest mean squared error, and the selected λ value is used to train the model on the entire data set. Due to the regularisation, only a small subset of variables at specific times will be included in the model and so the model will be easy to interpret and apply.

2.2.4 Validation

In order to assess how a model will perform on future batches, as well as for comparison of different models, it is vital to obtain a reasonable estimate of the prediction error. When data is plentiful, the prediction error can be estimated by setting aside a test set for the final model. However, in this case we have data sets of only 44 and 43 batches. If, say 8 batches are set aside as a test set, then first of all the model will suffer due to being trained on a significantly smaller training data set, and second of all the prediction error will be highly dependent on which batches are chosen for the test set, resulting in a poor estimate of the prediction error. In addition the choice of test set may arbitrarily favour one modelling approach over another. Another approach often seen, is to use the cross validation error from the parameter selection as a prediction error estimate. However, this estimate is biased, as the error is directly minimised for selecting the parameter [10]. To correctly account for overfitting from model selection, a separate cross validation should be performed to estimate prediction error [21]. For each fold, the entire model selection procedure is performed (including an inner cross validation for selecting parameters). In this way, the prediction error estimate accounts for any overfitting due to parameter selection.

2.3 Phase 2 Models

After applying the methods in the preceding section, a model is obtained that predicts the final harvest time of a batch using process data from the first phase of a batch. In this section we present a method for updating the predictions during the second phase of the process. The classical approach [1, 15] to making predictions online as a batch progresses is as follows:

1. Offline model fitting: The model is fitted using offline batch process data from historical batches. A global alignment is performed to synchronise the process data and obtain same length batches. The desired model is fitted.

2. Online application of model: As a new batch progresses, first a partial alignment is performed between the data so far obtained for the ongoing batch and the complete reference batch. This alignment estimates the corresponding point in the reference batch currently reached by the ongoing batch and synchronises the existing ongoing batch data to this leading portion of the reference batch. Next, the future variable trajectories of the ongoing batch are predicted in order to obtain the necessary input data to use the model. The model is then applied to the combined actual and predicted ongoing batch data to predict the response variable.

In the case we are considering the response variable is harvest time and so would be explicitly contained in the synchronised set of complete offline batches. Therefore, fitting a model to predict the harvest time based on this data would not be useful. We take a different approach by focusing on the online alignment with dynamic time warping and fitting a model directly to the online warping information.

2.3.1 Online Alignment

The procedure for implementing DTW alignment online was presented in [15]. The problem consists of aligning a partially complete batch to a complete reference batch. This entails that the fixed endpoint constraint in the standard DTW algorithm (the constraint that the endpoint of the query trajectory has to be aligned to the endpoint of the reference trajectory) must be abandoned. Instead, an "open-ended" alignment must be performed between the current

section of ongoing batch and the complete reference batch. The open-ended DTW algorithm synchronises the query to that leading portion of the reference which minimises the accumulated distance between them. The resulting alignment estimates which point has been reached in the reference batch based on events so far in the query batch. For each new observation that arrives, a new partial alignment is performed to re-align the ongoing batch to the reference batch. This means, that past alignments can be completely revised as new information comes in. For example, at $t_{batch} = 15$ minutes into the ongoing batch, online alignment may identify the corresponding reference time then obtained as $t_{\text{reference}} = 18$ (in which case the ongoing batch may be said to be "running fast" relative to the reference batch, having completed 18 minutes of the reference trajectory in only 15 minutes). However, when the next observation is measured at say $t_{batch} = 16$, it may occur that the new alignment identifies the current reference time as $t_{reference} = 11$, revising the preceding alignment. This instability in online alignment reflects the obvious difficulty of not knowing the future behaviour of the ongoing batch. Each online alignment is a "best guess" based on information so far. To try to limit the instability of online alignments, we incorporated the following adjustments to the basic algorithm.

1. **Alignment Variable** In the second phase of the process, Base Flow is the variable which most closely reflects the biological state. However, DTW tends to align according to magnitude of trajectories rather than shape, especially in the online version where there is no endpoint constraint. There is substantial variation between batches in the magnitudes of the Base Flow curves. Figure 3 shows an example where the ongoing batch has generally smaller Base Flow than the reference. When Base Flow is used as the alignment variable, it aligns according to the magnitude of Base Flow regardless of the smaller shape properties of the curves which are known to be more informative of the biological state (e.g. the small sinusoidal wave pattern that often occurs around two thirds of the way up the curves). To overcome this magnitude problem, Derivative Dynamic Time Warping (DDTW) was used [22]. In this version of DTW, the derivative of the trajectory is used as the alignment variable resulting in a more shape based rather than magnitude based alignment. To calculate the derivative of Base Flow, we first applied an exponentially weighted moving average to Base Flow (with smoothing factor $\alpha = 0.5$) to reduce noise. Letting x_i

denote the i^{th} sample of this smoothed Base Flow, then the derivative was estimated by:

$$\frac{d}{dt}(x_i) = \frac{1}{2}[x_i - x_{i-1} + \frac{1}{2}(x_i - x_{i-2})] \tag{6}$$

2. **Local Constraint** In the offline alignment of the phase 1 data a local constraint with $P = 1$ was found to be appropriate. Therefore, this same local constraint was applied for the online alignment during phase 2.

3. **Global Constraint** In the case of open ended alignment where the current end point of the ongoing batch is free to be matched to any point in the reference, it is advisable to implement a global constraint in addition to the local constraint. In DTW, a global constraint limits the region in which the warping function may be found, i.e., it specifies a limited range of reference times which may be matched to each ongoing batch time. Following [15] we calculate the global constraint upper and lower bounds from the empirical range of the warping functions resulting from an offline alignment of the phase 2 data for all batches (see Figure 4). In the offline alignments, the warping functions are guided by the future knowledge of the endpoint constraint and are therefore more accurate. Using the resulting global constraints for online alignment of a new batch thereby incorporates knowledge of how past batches evolved in order to improve the online alignment.

Applying the above modifications to the online DTW algorithm results in greater stability with regards to identification of the current reference time reached during the ongoing batch as illustrated in Figure 5.

An alternative approach to dealing with the instability of online DTW was devised by [23] who presented a "relaxed greedy DTW" algorithm which essentially only updates the warping function within some window around the current batch time, and fixes in place the warping function found before this window. This approach leads to far less variability in the online warping function improving the false alarm rate in their monitoring case study. In our case we prioritise finding the best alignment to the reference at each time point, and therefore do not apply the relaxed greedy DTW method which does not allow a bad alignment to be revised substantially in response to new data. We only wish to limit these revisions within reasonable bounds

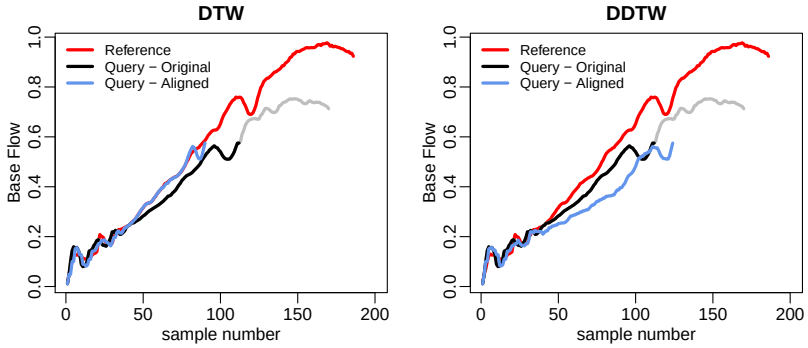

Figure 3: Online (open-ended) alignment using regular DTW (left) and using Derivative DTW (right) where the ongoing query batch is known up to sample 112. The future path of the query batch is shown in grey.

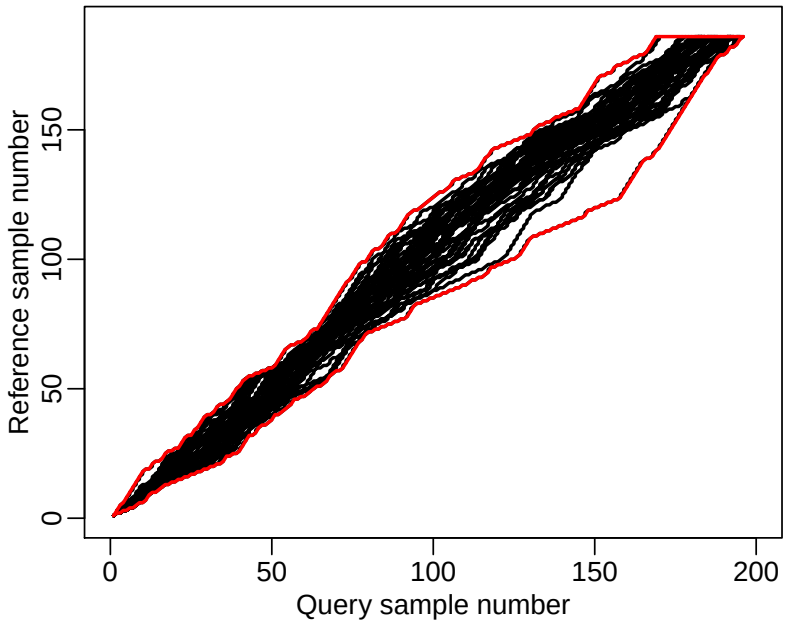

Figure 4: Calculation of global constraints to use for online alignment(red) as the empirical range of the warping functions (black) from an offline alignment of the data.

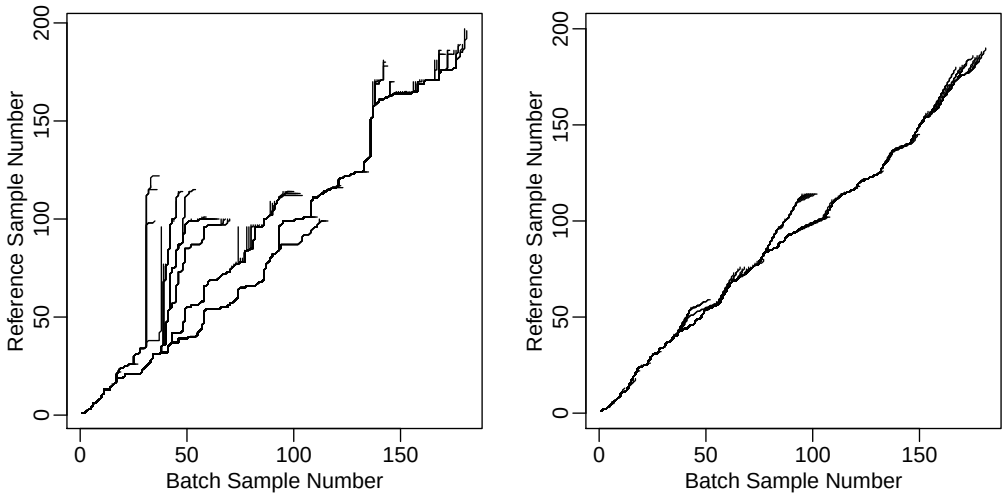

Figure 5: For a single batch, all online warping functions are shown. Left: using DDTW without any local or global constraints. Right: Using DDTW with local constraint $P = 1$ and empirical global constraints. These measures encourage less revisions in the warping functions as evident by the lesser degree of branching in the right hand plot

using the above safety measures, rather than remove the warping revisions completely.

2.3.2 Updating Predictions Online

Table 2: Online DTW Harvest Time Model Notation

Symbol	Definition
k	Current time (observation number) of the ongoing batch
$f(k)$	Current reference time according to online DTW
Δ_k	The time to reference batch ($= k - f(k)$)
$\overline{\Delta}_k$	Mean time to reference batch of all alignments up to time k
$\overline{\Delta}_y$	Mean time to reference batch of all alignments up to the end (harvest time) of the batch
$\hat{y}_{phase1}$	Predicted harvest time from lasso model at end of phase 1 of process
$\hat{y}_k$	Updated prediction at time k of harvest time using DTW
α	Weight of the phase 1 prediction in the online harvest time prediction
β	Weight of the mean time to reference batch in the online harvest time prediction

In this section we present how to combine the online DTW information with the phase 1 predictions. The notation used is summarised in Table 2. Let k denote the current observation number of the ongoing batch (i.e., the "real time"). Let $f(k)$ denote the corresponding current reference time determined from online DTW. Note that $f(k) \in \mathbb{R}$ and corresponds to the maximum reference time in the current online DTW warping function. Each online warping function must itself be monotonically increasing, but because the warping function is re-calculated for each new observation of the ongoing batch, f which we call the "outer warping function", does not have to be monotonically increasing. Let $\Delta_k = k - f(k)$ denote the "time to reference

batch". If Δ_k is negative, online DTW estimates that the batch is currently running faster than the reference batch, whilst positive Δ_k suggests that it is running slower. Next, we define the running average of time to reference batch as

$$\overline{\Delta}_k = \frac{1}{k}\sum_{l=1}^{k}\Delta_l \tag{7}$$

Hence, $\overline{\Delta}_k$ is a measure of how fast or slow the ongoing batch is relative to the reference batch on average up to time k. Finally, in order to combine the information provided by $\overline{\Delta}_k$ with the phase 1 predictions, we use the following linear model

$$\hat{y}_k = \alpha\hat{y}_{phase1} + \beta\overline{\Delta}_k \tag{8}$$

Equation (8) defines the online harvest time prediction at time k (denoted by $\hat{y}_k$) as the weighted sum of the prediction made at the end of phase 1 and the average time to reference so far. This means that if the ongoing batch progresses faster during the second phase, then $\overline{\Delta}_k$ will be negative and the phase 1 harvest time prediction will be reduced. To determine the precise amounts by which to weigh the phase 1 prediction and the online DTW information (denoted by the coefficients α and β respectively) the following steps are taken:

1. For all existing batches, perform online alignments throughout phase 2 and calculate the value of $\overline{\Delta}_y$ which is the mean time to reference of all alignments up to the end (harvest time) of the batch.

2. Regress y (the correct harvest time) against $\hat{y}_{phase1}$ (phase 1 harvest time prediction) and $\overline{\Delta}_y$ to obtain the least squares estimates of the coefficients α and β.

To apply the model on a new batch, the following steps are taken:

1. Upon completion of the first phase of the batch, perform offline alignment and apply the phase 1 lasso model to obtain the phase 1 harvest time prediction $\hat{y}_{phase1}$.

2. For each new observation during phase 2, perform online DTW alignment and calculate the value of $\overline{\Delta}_k$ then apply Equation (8) to obtain the updated harvest time prediction.

It is relevant to note that a harvest time prediction could be obtained from the online DTW alignment directly as, for example,

$$\hat{y}_k = y_{ref} + \overline{\Delta}_k \tag{9}$$

where the harvest time of the ongoing batch is estimated as the harvest time of the reference batch, y_{ref}, adjusted for the mean time to reference so far. For example, if the reference batch was harvested at $y_{ref} = 425$ and an ongoing batch is currently 10 time units faster than the reference batch on average ($\overline{\Delta}_k = -10$), then we predict the ongoing batch should be harvested at time $\hat{y}_k = 415$. We investigated this approach, but observed that even though the DTW information on its own provided a general indication of the speed of the ongoing batch relative to the reference batch, it was not powerful enough to provide accurate enough harvest time prediction. This was the reason for using the combined approach of Equation (8), which makes use of the DTW information according to the weight β. As will be seen in Section 3, in the case study we consider, the resulting values of α and β mean that the online predictions are dominated by the phase 1 prediction, plus only a small adjustment for the current mean time to reference. In other cases, if it is found that the online predictions are dominated by the DTW information (large β, small α), it may be worth considering the simplified version of Equation 9.

3 Results and Discussion

As the batches are not harvested consistently at the same point in the process, the first step was to cut off the data at the two harvest time criteria we defined. For product 1, this the data was cut off at the point where Base Flow falls to 95 % of its peak value (Figure 6) resulting in harvest times ranging from 395 to 461 time units. For product 2, the data was cut off according to the criterion that total Base Quantity reaches 0.72 in standardised units resulting in harvest times between 223 to 259 time units.

Next, each dataset was split into two according to the two phases of the process. The first phase was defined as the period from the start of the batch until the last observation with Base Flow equal to zero, with the first non-zero Base Flow sample onwards as the second phase.

DTW was used to align the phase 1 data as described in section 2.2.1, with the "dtw" package [24] in R . The alignment scores from the local constraint

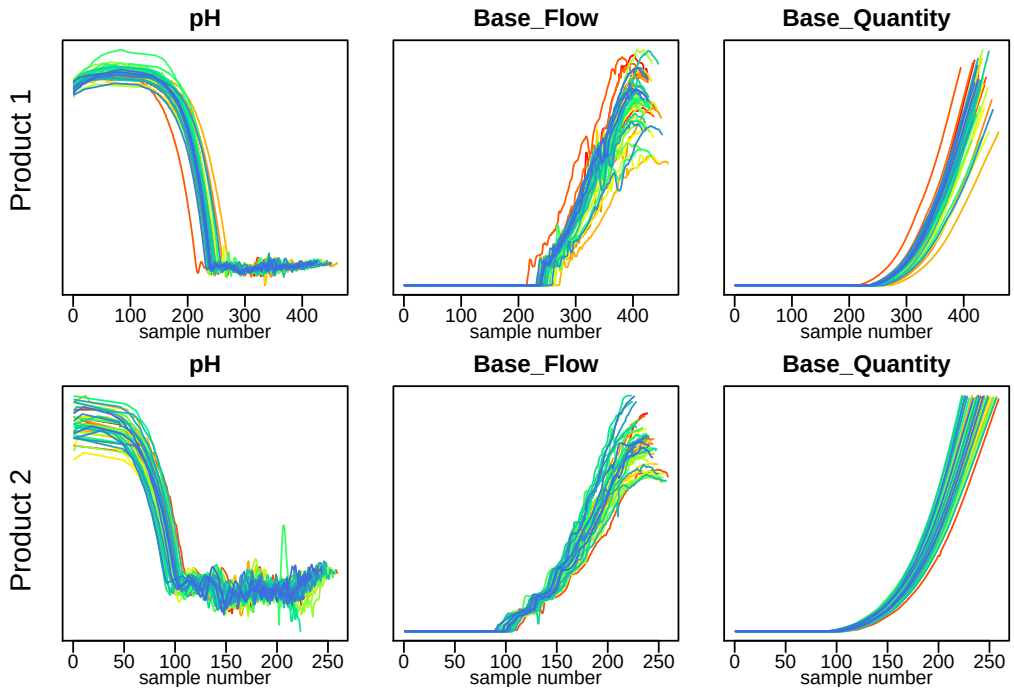

Figure 6: Data for product 1 cut off at the 95% base max harvest criterion and product 2 data after cutting the trajectories at the defined harvest time of Base Quantity = 0.72 in scaled units

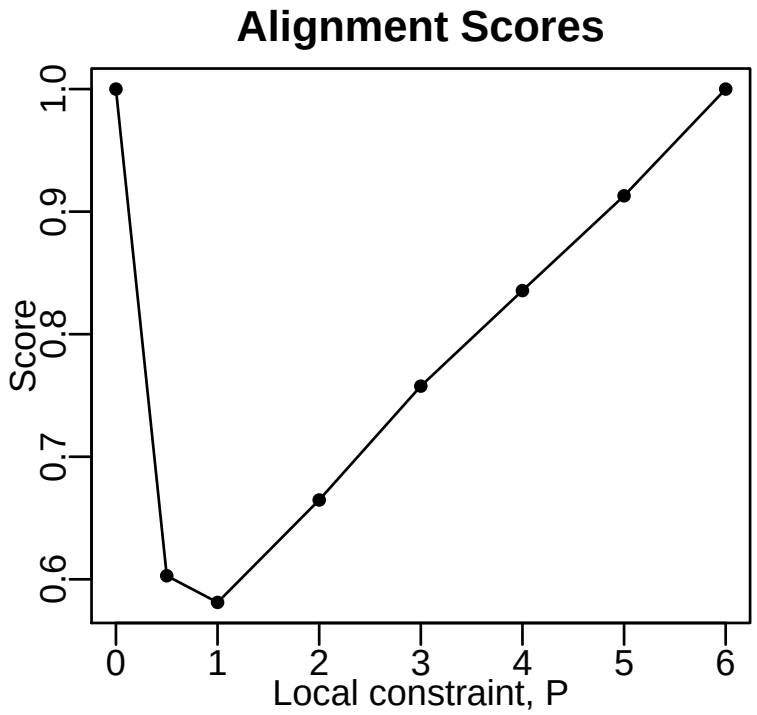

Figure 7: Alignment scores for selection of local constraint

selection procedure of [13] are shown in Figure 7 from which the $P = 1$ local constraint was chosen for product 1. The same result was obtained for product 2. Figure 8 shows phase 1 pH before and after alignment along with the DTW warping functions. After aligning the phase 1 data, every product 1 batch was 245 samples long, and every product 2 batch was 96 samples long. Then, for each product the aligned phase 1 data was unfolded. Only the pH variable and the warping function were included in the unfolding as Base Flow/Quantity are zero during phase 1. Thus the unfolded $\mathbf{X}$ matrix for product 1 was $I = 44$ batches by $J = 490$ "variables" (pH at reference times 1 to 245 and local batch time at reference times 1 to 245). The unfolded $\mathbf{X}$ matrix for product 2 was $I = 43$ batches by $J = 192$ "variables".

For each product, the unfolded matrix was auto-scaled by subtracting the mean from each column and dividing by the standard deviation. However, note that in subsequent cross validations for model fitting and RMSE estimation, auto-scaling was repeated using only the relevant training data. The

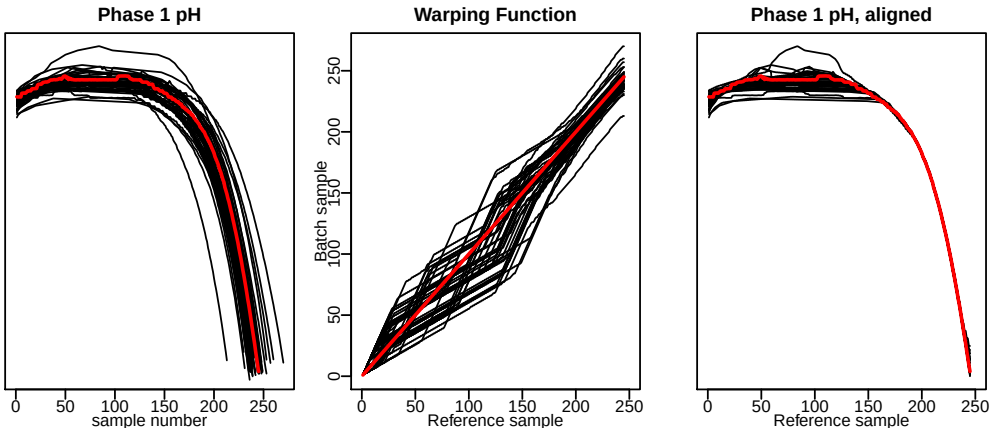

Figure 8: Phase 1 pH trajectories for product 1 before alignment (left) and after alignment (right) with the DTW warping functions (center)

response variable, $\mathbf{y}$ was centred but not scaled, so that RMSE calculations would be in original time units.

Using this processed phase 1 data, a Lasso model was fitted for each product as well as a PLS model as a benchmark (using the "glmnet" [25] and "pls" [26] R packages respectively). Hyper-parameters were selected using ten fold cross validation as shown in Figure 9 with $\lambda = 1.28$ and $n = 5$ resulting in smallest RMSE for product 1 and similarly $\lambda = 0.56$ and $n = 4$ for product 2. Using these parameter values the models are then fitted to the whole data sets resulting in 5 non-zero coefficients for the product 1 Lasso model and 10 non-zero coefficients for the product 2 Lasso model as shown in Figure 10. It is immediately seen that the main predictor of harvest time for product 1 was the batch time value at reference time 245 (in other words the duration of phase 1 as determined by the DTW alignment). However, the lasso model also selected pH at 5 critical times throughout phase 1. In contrast, the PLS model retained 5 components where each component includes a 450 element $\mathbf{X}$ loading vector (Figure 10). For product 2, the Lasso model is dominated by batch time at reference times 91 and 95, which again corresponds to the overall duration of phase 1 for product 2. The PLS model for product 2 consisted of 4 components. From these models the training RMSE was calculated as shown in Figure 11.

Although there was only a single informative variable during the first phase of the process, pH, it is still the case that this variable is measured

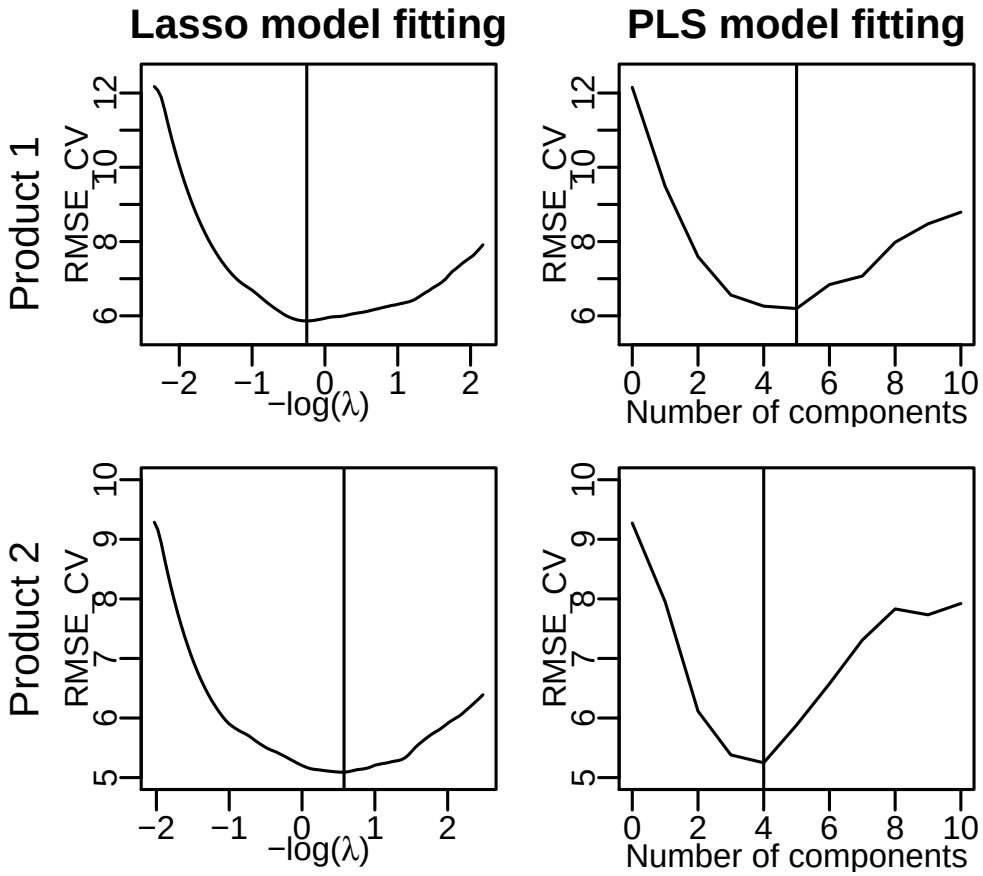

Figure 9: Cross validation results for selecting λ in the lasso model (left) and the number of components in the PLS model (right)

at many time points throughout the batch and the inclusion of warped time as a second variable brings the number of columns in the design matrix to 490 and 192 for products 1 and 2 respectively. Therefore, multivariate methods (lasso/PLS) are still appropriate. Using the knowledge gained from the lasso model, further refinement could be carried out to build an even simpler phase 1 regression model based on phase 1 duration and selected pH summary statistics. However, this would be a manual procedure, and we choose to focus on the automatic variable selection of the lasso method. Note that the PLS model alone gives no indication of the existence of a much simpler model structure for this data.

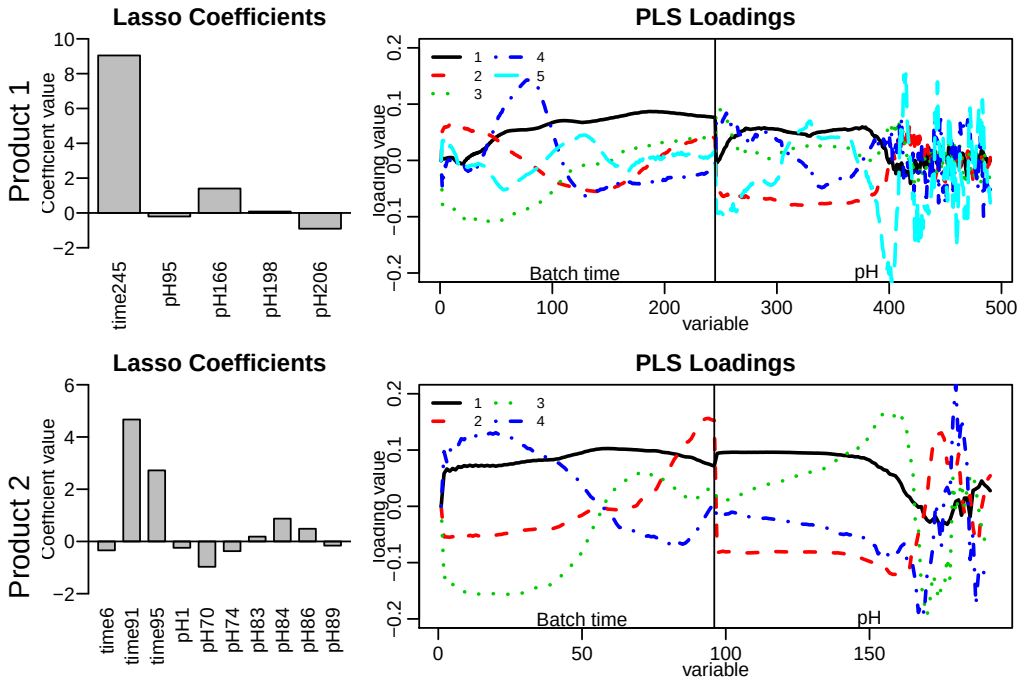

Figure 10: Coefficients of the lasso model and X loadings of the PLS model

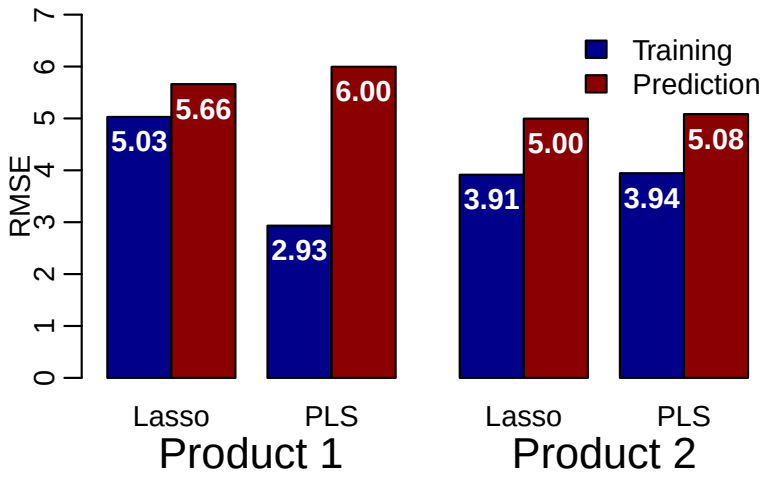

Figure 11: Training and prediction RMSE for the two phase 1 models for products 1 and 2

To estimate the prediction RMSE, a separate ten fold cross validation is performed. For each excluded fold, an inner ten fold cross validation is used to select the hyper-parameter so the error arising from each model fitting procedure is realistically estimated. The resulting estimated RMSE is shown in Figure 11. It was of interest that for both product 1 and product 2, the Lasso model resulted in the smallest prediction RMSE despite consisting of vastly fewer parameters than the PLS model. In addition, the Lasso model is easier to interpret. For these reasons, we chose the Lasso models for making predictions of harvest time at the end of phase 1.

The phase 1 lasso model provides a useful prediction of the harvest time upon completion of the first phase of the process. Next, we wished to update the predictions in real time during the second phase. Online alignments were calculated for the phase 2 data of each batch following the methods in Section 2.3.1. Then, for each batch f (the outer warping function), Δ_k (time to reference) and $\overline{\Delta}_k$ (running average time to reference) were obtained. For a single batch, Δ_k and $\overline{\Delta}_k$ are shown in Figure 12. Note that these quantities could be monitored directly by technicians to provide insight on the speed of an ongoing batch relative to the reference batch. In order to provide explicit harvest time predictions, the model described in Section 2.3.2 was fitted using the phase 1 Lasso predictions and the final values of $\overline{\Delta}_k$ of all

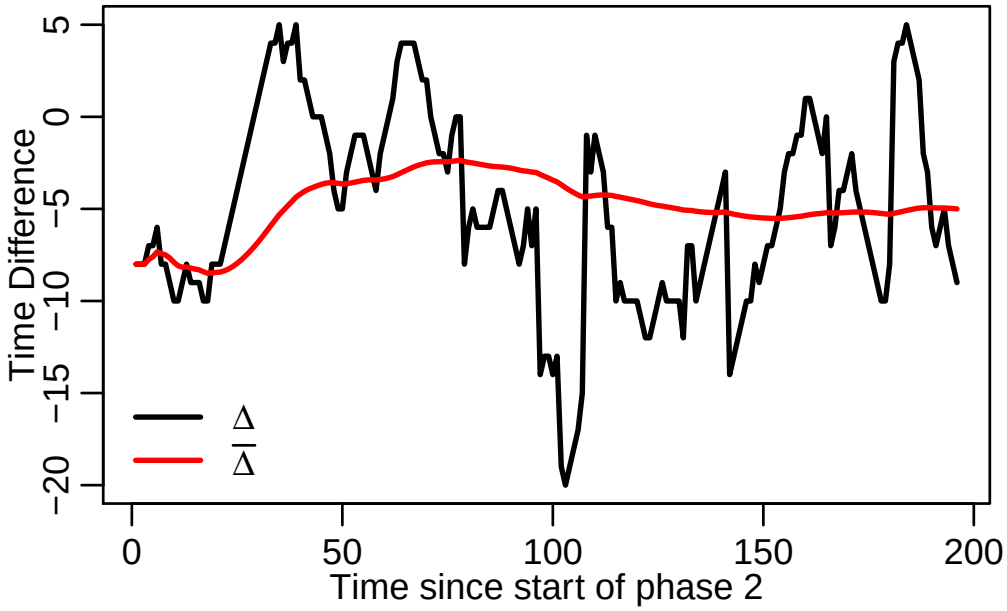

Figure 12: The time to reference (Δ_k) and running average time to reference ($\overline{\Delta}_k$) during phase 2 for a given batch

batches. This resulted in the coefficients given in Table 3. These coefficient values essentially mean that the online predictions calculated as the phase 1 prediction ($\alpha \approx 1$ for both products) with the addition of $\overline{\Delta}_k$ down-weighted according to β. This is logical because it means the predicted harvest time is increased when $\overline{\Delta}_k$ is positive and the ongoing batch is running slow on average during phase 2 compared to the reference batch.

Performance of the online model was evaluated based on the RMSE of the harvest time predictions across batches over time since the start of phase 2 (Figure 13). Prediction RMSE was estimated using ten fold cross validation. For each fold, the phase 1 lasso model was refitted as well as the online phase 2 model. The results show that both training and prediction RMSE decreases as phase 2 progresses (in both cases by over 0.2 in Product 1, and over 0.5 in product 2). Besides improving the prediction of harvest time, the process of online alignment is valuable for adding insight into the progress of the batch during phase 2.

Table 3: Phase 2 model coefficients

	α (p-value)	β (p-value)
Product 1	1.003 ($\sim$0)	0.177 (0.016)
Product 2	1.007 ($\sim$0)	0.343 (0.012)

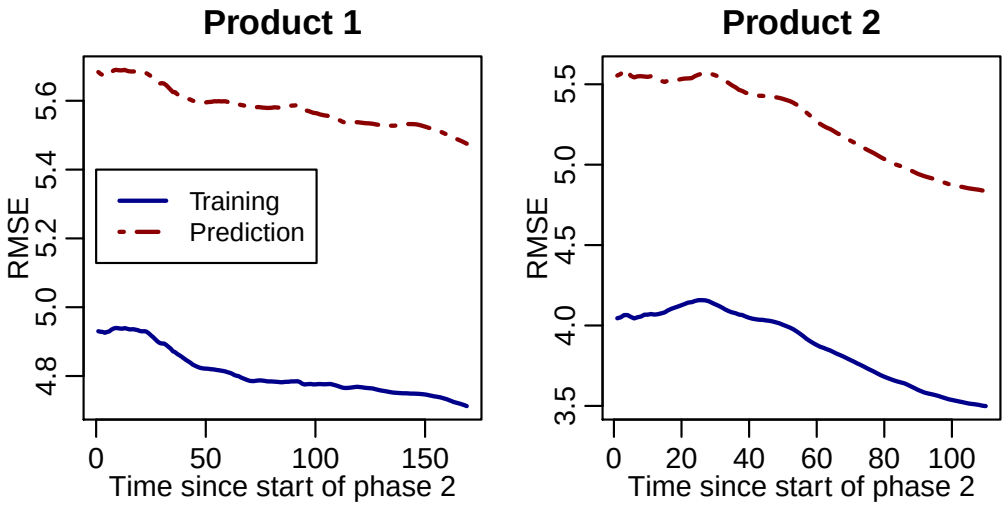

Figure 13: Online RMSE

4 Conclusions

We have presented a case study where a batch process exhibited considerable variation in its duration, and where batches were not harvested consistently according to some harvest criterion. There was a need for a method to predict the harvest time at an early stage in the process to facilitate resource allocation and ability to harvest consistently. We proposed using a Lasso regression model at the end of the first phase in the process and this model was shown to make useful predictions for two different products. We compared the Lasso model to a PLS model (the default approach in batch process prediction), and the Lasso model was shown to be more parsimonious and easier to interpret whilst performing as well as the PLS model. The Lasso model provides useful predictions of batch harvest time at an early stage in the process. Furthermore, we presented a novel method for updating the harvest time predictions during the second phase of the process by using information from online DTW alignment. This method was shown to reduce the RMSE of the predictions during the second phase. The proposed methods may be readily applied to other batch processes to anticipate harvest time and thereby contribute to improved process efficiency and quality.

5 Acknowledgements

The research is partially funded by BIOPRO (www.biopro.nu) which is financed by the European Regional Development Fund (ERDF), Region Zealand (Denmark) and BIOPRO partners. We thank Chr. Hansen A/S for providing access to data and expertise on the production process.

References

[1] Paul Nomikos and John F. MacGregor. Multi-way partial least squares in monitoring batch processes. *Chemometrics and Intelligent Laboratory Systems*, 30(1):97–108, 1995. ISSN 01697439. doi: 10.1016/0169-7439(95)00043-7.

[2] Cenk Ündey, Sinem Ertunc, and Ali Çinar. Online Batch/Fed-Batch Process Performance Monitoring, Quality Prediction, and Variable-Contribution Analysis for Diagnosis. *Industrial & Engineering Chem-*

istry Research, 42(20):4645–4658, 2003. ISSN 0888-5885. doi: 10.1021/ie0208218.

[3] David Wang. Robust data-driven modeling approach for real-time final product quality prediction in batch process operation. *IEEE Transactions on Industrial Informatics*, 7(2):371–377, 2011. ISSN 15513203. doi: 10.1109/TII.2010.2103401.

[4] Jon C. Gunther, Jeremy S. Conner, and Dale E. Seborg. Process monitoring and quality variable prediction utilizing PLS in industrial fed-batch cell culture. *Journal of Process Control*, 19(5):914–921, 2009. ISSN 09591524. doi: 10.1016/j.jprocont.2008.11.007.

[5] Barry Lennox, Gary A Montague, Andy M Frith, Chris Gent, and Vic Bevan. Industrial application of neural networks - an investigation. *Journal of Process Control*, 11(5):497–507, 2001. ISSN 09591524. doi: 10.1016/S0959-1524(00)00027-5.

[6] Kiran Desai, Yogesh Badhe, Sanjeev S. Tambe, and Bhaskar D. Kulkarni. Soft-sensor development for fed-batch bioreactors using support vector regression. *Biochemical Engineering Journal*, 27(3):225–239, 2006. ISSN 1369703X. doi: 10.1016/j.bej.2005.08.002.

[7] Zhengbing Yan, Chih-chiun Chiu, Weiwei Dong, and Yuan Yao. A LASSO-based batch process modeling and end-product quality prediction method. In *IFAC Proceedings Volumes*, volume 47, pages 6704–6709. IFAC, 2014. ISBN 9783902823625. doi: 10.3182/20140824-6-ZA-1003.00204.

[8] Ognjen Marjanovic, Barry Lennox, David Sandoz, Keith Smith, and Milton Crofts. Real-time monitoring of an industrial batch process. *Computers & Chemical Engineering*, 30(1012):1476–1481, 2006. ISSN 0098-1354. doi: 10.1016/j.compchemeng.2006.05.040.

[9] E. Latrille, G. Corrieu, and J. Thibault. pH prediction and final fermentation time determination in lactic acid batch fermentations. *Computers and Chemical Engineering*, 17:S423–S428, 1993. ISSN 00981354. doi: 10.1016/0098-1354(93)80261-K.

[10] Gavin C. Cawley and Nicola L. C. Talbot. On Over-fitting in Model Selection and Subsequent Selection Bias in Performance Evaluation. *Journal of Machine Learning Research*, 11:20792107, 2010. ISSN 1532-4435.

[11] Cenk Ündey and Ali Çinar. Statistical Monitoring of Multistage, Multiphase Batch Processes. *IEEE Control Systems*, 22(5):40–52, 2002. ISSN 1066033X. doi: 10.1109/MCS.2002.1035216.

[12] Salvador García-Muñoz, Mark Polizzi, Andrew Prpich, Cathal Strain, Adam Lalonde, and Vilmary Negron. Experiences in batch trajectory alignment for pharmaceutical process improvement through multivariate latent variable modelling. *Journal of Process Control*, 21(10):1370–1377, 2011. ISSN 09591524. doi: 10.1016/j.jprocont.2011.07.013.

[13] Max Spooner, David Kold, and Murat Kulahci. Selecting local constraint for alignment of batch process data with dynamic time warping. *Chemometrics and Intelligent Laboratory Systems*, 167(February):161–170, 2017. ISSN 18733239. doi: 10.1016/j.chemolab.2017.05.019.

[14] Hiroaki Sakoe and Seibi Chiba. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49, 1978. ISSN 00963518. doi: 10.1109/TASSP.1978.1163055.

[15] Athanassios Kassidas, John F. MacGregor, and Paul A. Taylor. Synchronization of batch trajectories using dynamic time warping. *AIChE Journal*, 44(4):864–875, 1998. ISSN 00011541. doi: 10.1002/aic.690440412.

[16] Henk J. Ramaker, E. N M Van Sprang, Johan A. Westerhuis, and Age K. Smilde. Dynamic time warping of spectroscopic BATCH data. *Analytica Chimica Acta*, 498(1-2):133–153, 2003. ISSN 00032670. doi: 10.1016/j.aca.2003.08.045.

[17] Abdullah Mueen and Eamonn Keogh. Extracting Optimal Performance from Dynamic Time Warping. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, pages 2129–2130. ACM Press, 2016. ISBN 9781450342322. doi: 10.1145/2939672.2945383.

[18] Paul Geladi and Bruce R. Kowalski. Partial least-squares regression: a tutorial. *Analytica Chimica Acta*, 185(C):1–17, 1986. ISSN 00032670. doi: 10.1016/0003-2670(86)80028-9.

[19] Changliang Zou, Wei Jiang, and Fugee Tsung. A LASSO-Based Diagnostic Framework for Multivariate Statistical Process Control. *Technometrics*, 53(3):297–309, 2011. ISSN 0040-1706. doi: 10.1198/TECH.2011.10034.

[20] Robert Tibshirani. Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58 (1):267–288, 1996.

[21] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*, volume 1. Springer New York, 2001. ISBN 978-0-387-84857-0. doi: 10.1007/b94608.

[22] Eamonn J Keogh and Michael J Pazzani. Derivative Dynamic Time Warping. In *First SIAM International Conference On Data Mining (SDM'2001*, pages 1–11, 2001. ISBN 978-0-89871-495-1. doi: 10.1137/1.9781611972719.1.

[23] J. M. González-Martínez, Alberto Ferrer, and Johan A. Westerhuis. Real-time synchronization of batch trajectories for on-line multivariate statistical process control using Dynamic Time Warping. *Chemometrics and Intelligent Laboratory Systems*, 105(2):195–206, 2011. ISSN 01697439. doi: 10.1016/j.chemolab.2011.01.003.

[24] Toni Giorgino. Computing and Visualizing Dynamic Time Warping Alignments in R : The dtw Package. *Journal of Statistical Software*, 31 (7):1–24, 2009. ISSN 1548-7660. doi: 10.18637/jss.v031.i07.

[25] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software*, 33(1):1–20, 2010. ISSN 1548-7660. doi: 10.18637/jss.v033.i01.

[26] Bjørn-Helge Mevik, Ron Wehrens, and Kristian Hovde Liland. *pls: Partial Least Squares and Principal Component Regression*, 2015. URL `http://cran.r-project.org/package=pls`.

CHAPTER 5
Predicting pectin quality with CP Kelco

5.1 Introduction

CP Kelco, Lille Skensved is the largest pectin factory in the world (Kelco 2015) and produces pectin for use in the food and pharmaceutical industries. Besides its long history as a gelling agent in jams and preserves, pectin is also used, for example, to improve the mouth-feel of low fat dairy products, and the skin-feel of cosmetics. Pectin is a complex polysaccharide synthesised by plants, and present in all plant cell walls (Flutto and Danisco 2003). At CP Kelco, the pectin is extracted from dried citrus peel which is imported from South and Central America and Southern Europe. This raw pectin is then put through a sequence of steps to transform it into one of several pectin products that meet the specifications required for a particular application. Pectin is made up of chains of galacturonic acid units. Each unit may be altered by either de-esterification or amidation, and two important parameters characterising pectin are

- Degree of esterification (DE): The percentage of total number of galacturonic acid units that are esterified
- Degree of amidation (DA): The percentage of total number of galacturonic acid units that are amidated

Fig. 5.1 illustrates how DA and DE relate to the structure of the pectin molecule. In general, only those galacturonic units in the pectin molecule which are esterified may be converted to amidated units, so as DA increases, DE decreases and the two

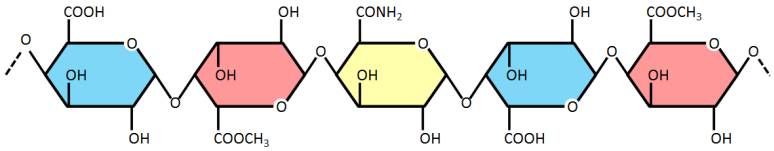

Figure 5.1: Chain of five galacturonic acid units in a pectin molecule. The red units are naturally esterified, and the yellow unit is amidated. Thus, for this portion of the molecule, DE = 40 % and DA = 20 %.

parameters are negatively correlated. However, as there are many other factors influencing the values of DA and DE, they are not perfectly correlated. Note also that Fig. 5.1 is a simplified diagram of one part of a pectin molecule. The complete pectin molecule is vastly more complex and may contain branching of the neutral sugar chains and a variety of chemical components. The structure and composition of the pectin molecule varies greatly depending on its biological origin.

The values of DE and DA determine properties such as gelling strength, speed, and optimal pH and temperature for gelling. In the food industry, pectin is classified into three broad categories, High ester pectin (DE $>$ 50 %, DA $\approx$ 0%), low-ester conventional pectin (DE $<$ 50 %, DA $\approx$ 0 %), and low-ester amidated pectin (DE $<$ 50 %, DA $<$ 25 %).

The project with CP Kelco focused on the amidation stage in their production of low-ester amidated pectin. This stage consisted of the following steps

1. A quantity of raw pectin is placed in one of the amidation reactors
2. The pectin is mixed with a solution of ammonia-alcohol under controlled conditions (temperature, pH, ammonia concentration, duration etc.)
3. The ammonia is drained from the reactor
4. The now-amidated pectin is rinsed with alcohol
5. The alcohol is drained from the reactor

It should be noted that there are many steps, both before (such as extraction and purification of pectin from the citrus peel), and after (such as drying and grinding), the amidation stage of the process, but these parts of the process were not within the scope of the project.

After the amidation phase and subsequent steps, groups of around 5 to 8 amidated pectin batches from the reactors are mixed to form semi-finished products. Fig. 5.2

shows a simplified diagram of the production set up from which the data was obtained. Many variables are measured throughout each batch, both in the particular reactor where each batch is treated, and in Tank 2 which supplies all of the reactors with ammonia and receives waste ammonia (weakened by the amidation process) from the reactors. An automatic control mechanism keeps ammonia concentration at the desired level in Tank 2. We refer to the variables measured in tank 2 and the reactors as the process variables and they include, e.g., temperatures, concentrations, pressures and flow rates at different points in the system. In addition, DE of the ingoing raw pectin is measured intermittently (before it is separated into different reactors), and this measurement is referred to as precursor DE. Depending on the project phase, the response data consisted of DE and DA measurements for a subset of the batches leaving the reactors, or of DE and DA measured on the semi-finished products.

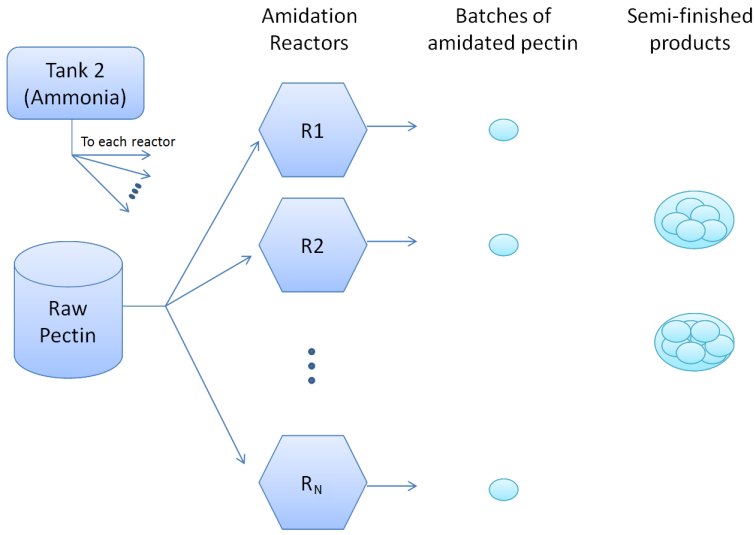

Figure 5.2: Simplified diagram of production set up.

The project consisted of two phases, each using a different dataset. In the first phase of the project, the original goal was to predict DA/DE of the semi-finished products using the process variables. These predictions could then be used by operators to select which batches to combine into a semi-finished product without directly measuring DA and DE of every batch. However, this was soon found to be unrealistic as the association between process data for a single batch, and the quality data measured on a semi finished product which consists of that batch combined with several other batches, was found to be too weak. Therefore, the goal was revised to predicting DA and DE of the batches directly. Due to the high dimensional and high correlation structure of the input data, lasso regression was chosen to fit the

prediction model. The dataset considered for this phase consisted of 274 batches, of which quality parameters DA and DE were known for 92 batches. The fitted model performed poorly at predicting DA and DE. The main reason for this was likely that all batches in the dataset were from the same production run of pectin for the same product with product specifications consisting of a narrow range of DA and DE values. Therefore, there was too little structured variation in the response data for the model to predict. It was decided to initiate a second project phase based on a new dataset.

The goal for the second project phase was the same, that of predicting DA and DE of the batches using process data. However, this time the data was collected from a production run that covered a variety of products with greater variation in product specifications of DA and DE. Again, lasso regression was applied and the fitted model was able to predict DA and DE with greater success than previously. In addition, the variables selected by the lasso model could be interpreted in terms of the theory behind the process, and led to greater process understanding. The model was compared to a partial least squares prediction model and equivalent performance was observed, despite the substantially simpler structure of the lasso model.

In this chapter, results from the second phase of the project with CP Kelco are presented. In the following sections, issues related to data pre-processing are discussed, and an exploratory analysis of the data is performed. The lasso model is presented and interpreted, and contrasted with a benchmark PLS model. Finally, the chapter concludes with a discussion of the lessons learned from this project in relation to the general context of batch process prediction.

5.2 Data preprocessing and exploratory data analysis

5.2.1 Process data

The raw process data was obtained in the form of a single time series for each variable from each of the eight reactors and tank 2. There were 17 numerical variables measured by sensors during each batch, as listed in Table 5.1. Note that the first 11 variables are repeated in each of the reactors, whilst the last 6 variables are measured in Tank 2 which is a single location (see Fig. 5.2). Besides the numerical variables, a categorical step-code variable indicated in which process step each reactor was engaged at any time.

Batch identifier information was also provided, indicating the batch-number, start-time, and reactor number of each batch making it possible to extract the raw process data belonging to each individual batch. As is common for industrial data, the sampling frequency of the raw data varied for different variables. For some variables values were present every few minutes, whilst others they were saved every few mil-

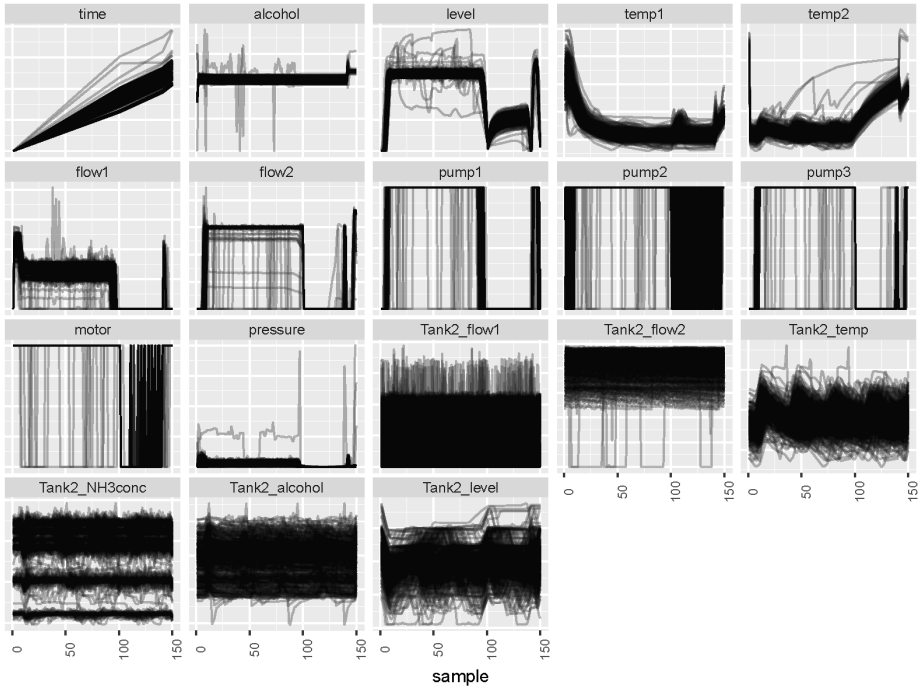

Figure 5.3: Process variable trajectories for all batches in dataset.

liseconds. Linear interpolation was used to obtain values at the same time points for all variables. The duration of each batch varied, so sampling the variables at a fixed time-frequency would result in different numbers of observations for different batches. Of course, alignment techniques could then be applied to address this. However, it was possible to avoid the issue of alignment completely by making use of the detailed step-code information. The 17 process variables were sampled at 150 points throughout each batch as follows:

- 100 samples evenly spaced during amidation phase
- 40 samples evenly spaced during ammonia draining phase
- 10 samples evenly spaced during rinsing in alcohol phase

This results in data that is synchronised according to the important steps in the process, i.e., the 120th sample for a batch always corresponds to around the midpoint of the ammonia draining phase. The time since the beginning of each batch

Table 5.1: Variables measured during the amidation process.

Variable	Description
alcohol	Alcohol concentration at reactor
level	Level in reactor
temp1 temp2	Temperature measured at two locations around reactor
flow1 flow2	Flow rate at two locations around reactor
pump1 pump2 pump3 motor	Binary (on/off) indicator for three pumps and a motor
pressure	Pressure at reactor
Tank2_flow1 Tank2_flow2	Flow rate at two locations around Tank 2
Tank2_temp	Temperature in Tank 2
Tank2_NH3conc	Ammonia concentration in Tank 2
Tank2_alcohol	Alcohol concentration in Tank 2
Tank2_level	Level in Tank 2

was calculated as an additional variable, as this information is not contained in the sample number from this sampling procedure.

The process data contained 468 batches. The 17 numerical process variables and time variable are plotted for all 468 batches in Fig. 5.3. The three stages in the process are most clearly reflected in the level variable. The level rises as the reactor is filled with ammonia, remains high during amidation phase up to sample 100, then drops as ammonia is drained. At samples 100 to 140 the wet pectin continues to drip excess ammonia so level rises slightly. This is drained and finally the level rises as the pectin is washed in alcohol during samples 141 to 150. Note that the y-axis scales are omitted for confidentiality reasons.

As a preliminary exploration of the process data, multiway PCA was applied. The four binary variables were not included in this analysis. A 468 x 2100 data matrix was constructed by placing the 150 samples of the 14 continuous process variables side by side for each batch. The matrix was centered and scaled (Fig. 5.4), and PCA was applied. From visual inspection of the scree plot of the eigenvalues (Fig. 5.5), 10 components were retained in the model explaining 68.7 % of the variance.

The Hotelling's T^2 and Q statistics from the 10 component model were calculated for each batch (Fig. 5.6) and atypical batches were identified as the 5 % of batches with greatest T^2 value. These batches are shown in Fig. 5.7. This shows that T^2 has

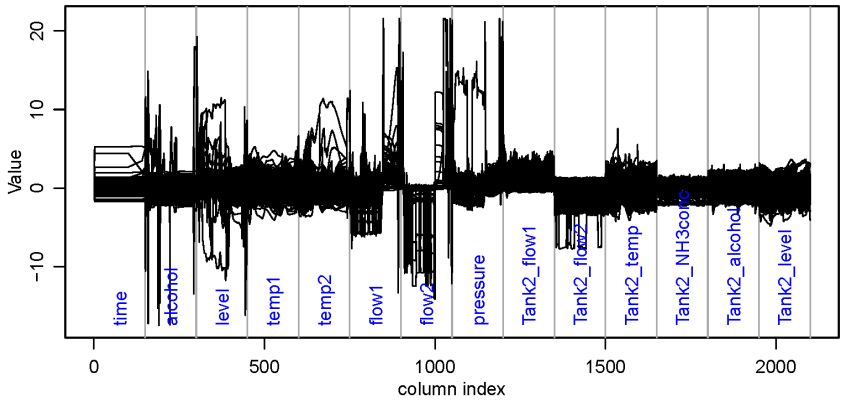

Figure 5.4: The unfolded and scaled data for all batches.

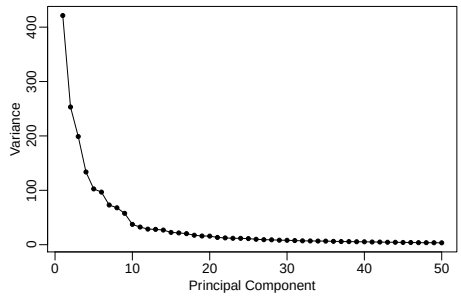

Figure 5.5: Scree plot of the eigenvalues from PCA analysis.

identified unusual variable trajectories. Reasons for the behaviour of these batches were not found, and so it may not be appropriate to consider them outliers. In this work we do not exclude them from model fitting and prediction error estimation.

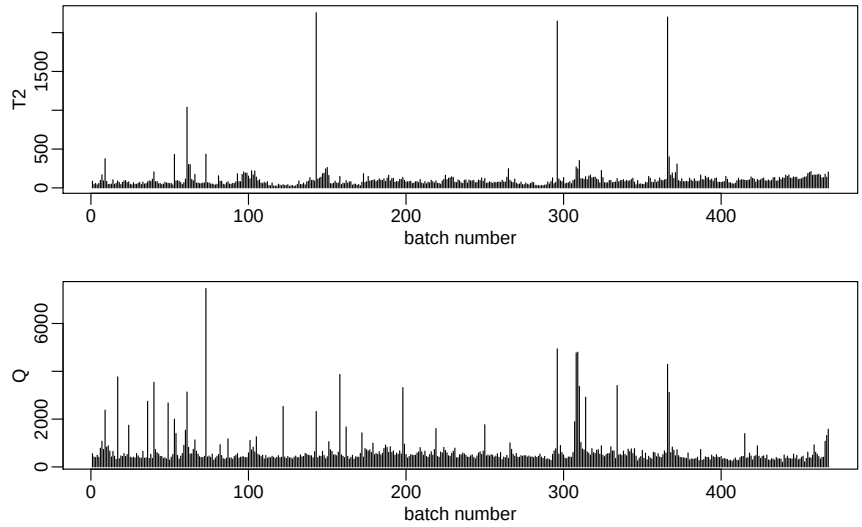

Figure 5.6: Hotelling's T^2 and Q statistics for all batches.

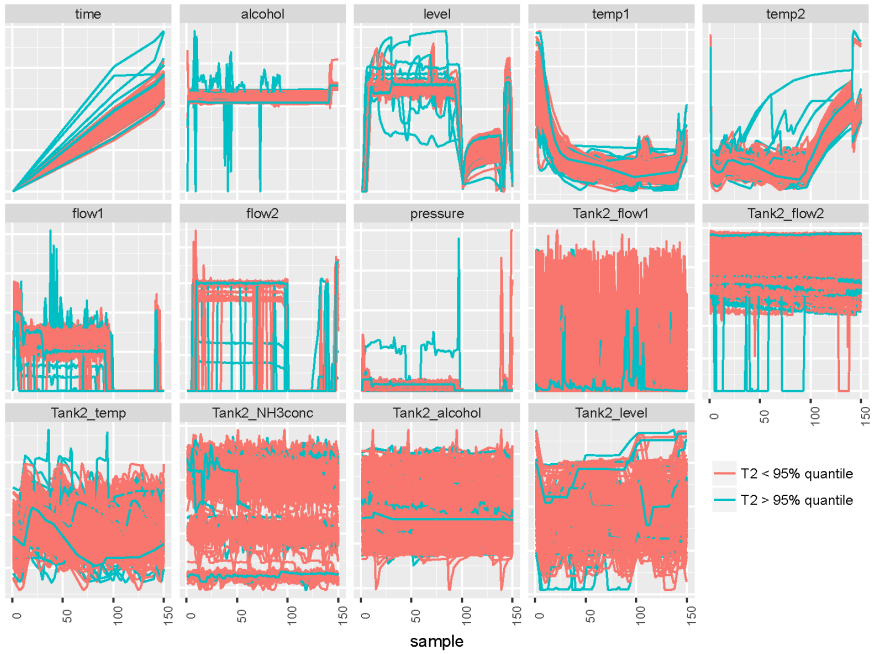

Figure 5.7: Those batches with extreme Hotelling T^2 values (green) and those with non-extreme values (red).

5.2.2 Batch background data

Besides the process variables, there was also some background data regarding each batch, consisting of the set-points (target values) of certain process conditions, and the measured value of DE of the raw pectin:

- NH3_SP: Tank2 Ammonia concentration set-point (6 different levels)
- Temp_Tank2_SP: Tank 2 Temperature set-point (2 levels)
- pH_SP: pH set-point (6 levels)
- Precursor DE (PreDE): The measured DE of the raw pectin before it enters the reactors

In rare cases a set-point could change mid-way through a batch, but in general the set-point for each condition was constant throughout a batch. Therefore, a single value was extracted for each set-point condition per batch (applying majority rule if a set-point changed). The different set-points for ammonia concentration are reflected in the ammonia concentration actually measured (Tank2_NH3conc shown in in Fig. 5.3), where the lower two bands are batches with the lowest two levels of ammonia set-point and the thicker upper band corresponds to batches with the highest 4 levels of ammonia set-point.

Finally, precursor DE was considered a highly relevant input variable as it was expected the final DA and DE values depended somewhat on the DE of the raw pectin going into the reactors: if DE is high before amidation, then DE and DA is expected to be higher than usual after amidation. Precursor DE was measured intermittently, with clusters of measurement repetitions separated by periods of non-measurement. As precursor DE was measured less frequently than the start-times of batches some processing was required to obtain a single PreDE value per batch. The available precursor DE data was smoothed using local regression (the R loess function), with a span determined by cross validation. The raw and smoothed PreDE data is shown in Fig. 5.8. Then the loess model was used to infer the value of precursor DE at the start time of each batch.

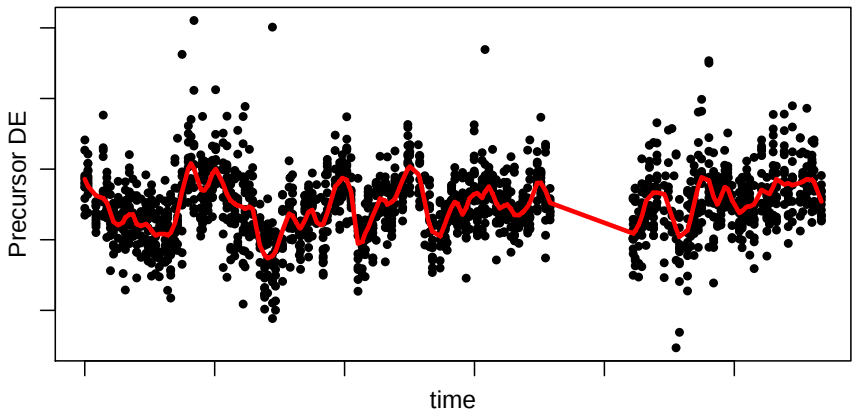

Figure 5.8: The raw and loess-smoothed precursor DE measurements. The gap in measurements corresponds to a period where no batches were processed, so is of no consequence for model fitting.

5.2.3 Response data

The response data consisted of DA and DE measured on the completed pectin batches using infrared (IR) spectroscopy. For each batch, the response variables were calculated from the mean of three IR measurement repetitions, so inference regarding measurement system uncertainty was possible. For DA, the estimated standard error of the mean of the three IR measurements was 0.05 (in scaled units), whilst for DE it was 0.08. This provides an estimate of the measurement system repeatability alone (variation of measurements taken under the same conditions within a short period of time) which is only one component of measurement error. However, the standard errors provide a lower bound for measurement error of the response variables and thereby also a lower bound for the prediction accuracy we may hope to achieve.

After mean-centering and scaling by the standard deviation, DA and DE are shown in Fig. 5.9. The negative correlation between DA and DE is clearly reflected in the measurements.

5.3 Results and discussion

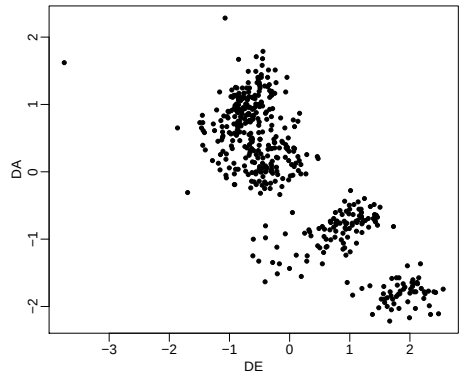

Figure 5.9: The centred and scaled response variables DA and DE for all batches.

5.3.1 Model fitting

To fit the models, the process data was unfolded as described in Section 5.2.1. Four additional columns were appended to the data matrix consisting of the three set-point variables, and precursor DE. The data matrix was centred and scaled.

Four models were considered:

- Lasso model for DA
- Lasso model for DE
- PLS model for DA
- PLS model for DE

Ten-fold cross validation and the one standard error rule was used to select the hyper-parameters of each model (lambda for the lasso models, number of components for the PLS models). The results of this cross validation are shown in Fig. 5.10. The models were then refit using the complete dataset and the selected hyper-parameter values. The fitted lasso models for DA and DE then had 24 and 16 non-zero coefficients respectively, and the coefficient values are shown in Fig. 5.11. The standard errors of the coefficients were estimated using the bootstrapping approach suggested by Tibshirani (1996). This procedure consists of re-sampling the dataset with replacement, fitting a lasso model with the chosen lambda value, and saving the resulting bootstrap coefficient estimates. After repeating many times (in this case 1000), the standard errors of the bootstrap coefficient values are calculated. As the Lasso models contain a limited number of non-zero coefficients, they can quite easily be interpreted.

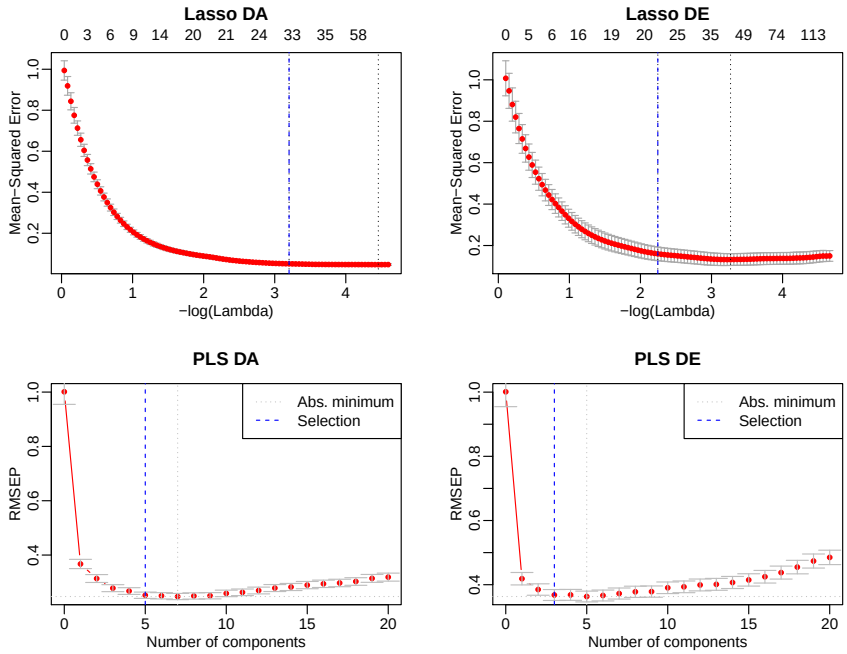

Figure 5.10: Cross validation results for selecting the model hyper-parameters. For each plot model complexity increases along the x axis. The numbers above the lasso plots indicate the number of non-zero coefficents in the model for the corresponding lambda value.

The DA model included positive coefficients for time at sample 150 (i.e. the overall duration of the batch), Tank2_NH3conc at 19 sampling points throughout the batch and PreDE. This agrees with prior knowledge that greater duration, precursor DE and ammonia concentration should lead to greater DA. For the DE model, negative coefficients were found for time at sampling points near the end of the batch, and various sampling points for alcohol and ammonia concentration. A positive coefficient for precursor DE was found. This also reflects the theory behind pectin amidation.

For the PLS models, cross validation results implied that 5 components and 3 components were appropriate for predicting DA and DE respectively. The coefficients of the PLS models are shown in Fig. 5.12. Each PLS model contains 2104 non-zero coefficients. Similar effects of time, ammonia concentration and precursor DE discsused above can be observed, but they are somewhat obscured by the many other non-zero coefficients.

The observed response variables are plotted against the values predicted by each

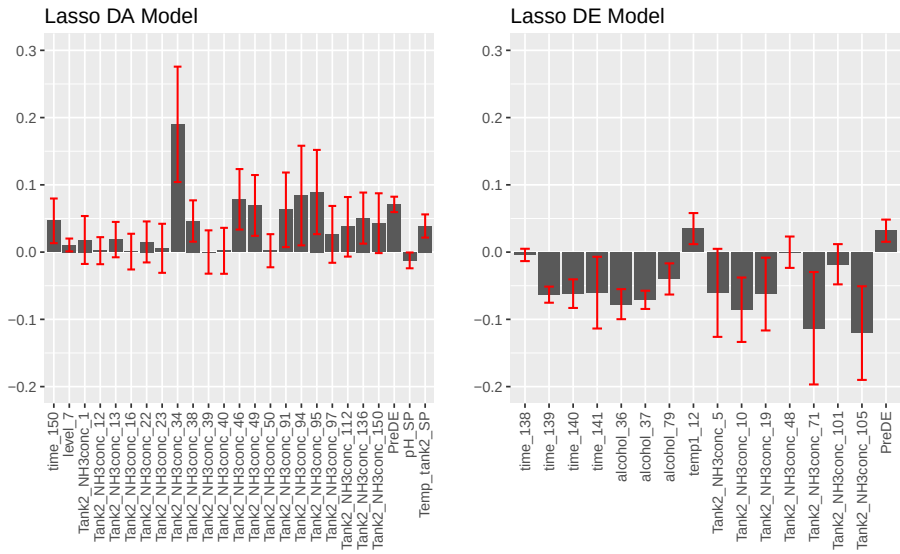

Figure 5.11: The non-zero coefficients of the lasso models, ± 1 standard error bars in red.

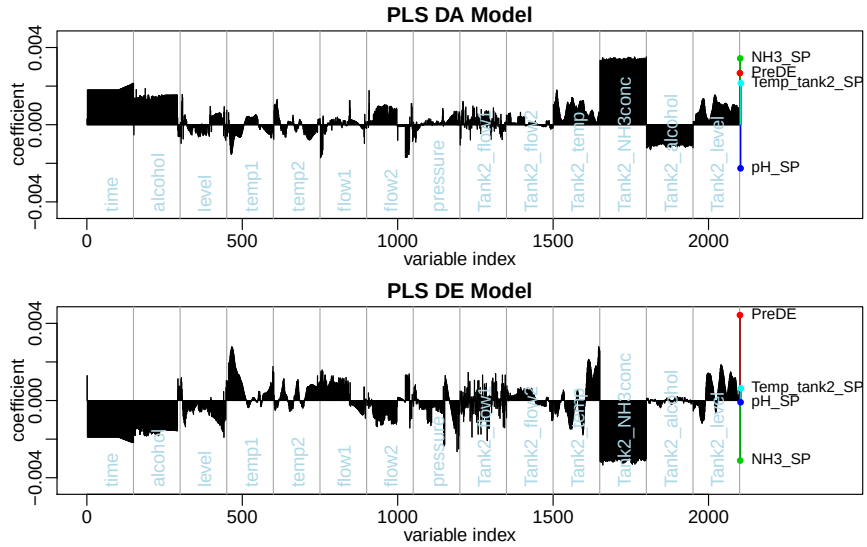

Figure 5.12: Coefficients for the PLS models.

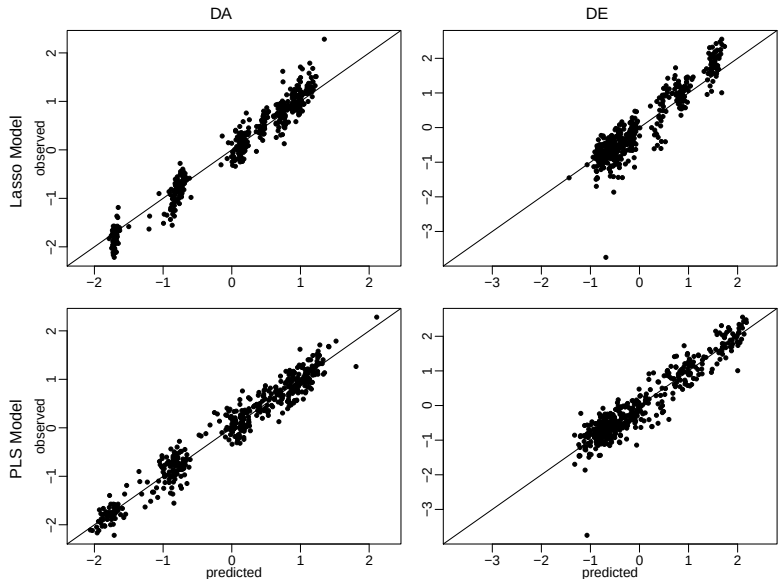

Figure 5.13: Observed against predicted values for the four models.

model in Fig. 5.13. In this context, the PLS models appear slightly better. However, these predictions are for the training data. For a more reliable validation of model performance, the prediction error for unseen data must be estimated.

The lasso model has slightly better RMSE than the PLS model for predicting DA. For predicting DE, PLS clearly has the smaller prediction error. However, all the error estimates are rather large considering the unit variance of the scaled response variables.

5.3.2 Estimation of prediction error

In order to evaluate the usefulness of the models, an estimate of their prediction error when used on new data must be made. Ten fold cross validation was used to obtain this estimate. For each fold, the entire model fitting procedure was performed (including selection of model hyper-parameters using an inner cross validation loop). Then the models were used to predict DA and DE of the left-out fold and the prediction RMSE obtained. The mean of the RMSE over the ten folds is then a good estimate of the RMSE of the models if they are applied to new data. These estimates are shown in Fig. 5.14.

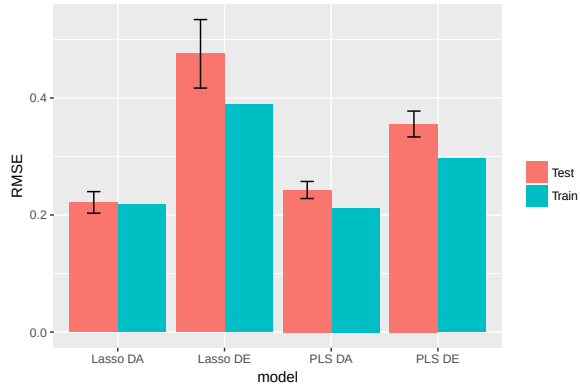

Figure 5.14: Estimated prediction (test) RMSE of the four models from ten fold cross validation, ± one standard error (black bars). Train RMSE was calculated using the final models and the complete data.

5.4 Chapter conclusion

In this project methods for predicting the quality parameters DA and DE using batch process data for pectin amidation were investigated. Lasso regression was used to make predictions using only a small subset of the many variables in the input data, and this lead to easily interpretable models and confirmed the hypothesis that precursor DE was an important variable for final DA and DE. The lasso models were compared to PLS models, and for predicting DA they had a similar accuracy. For predicting DE, the PLS model had smaller RMSE. However, RMSE of the predictions was too large in all cases for the models to be used for quality prediction in future pectin production at CP Kelco. It is likely that model performance was limited by the measurement uncertainty of the response variables.

Data taken from routine production was used in this project. This had the advantage that a large amount of batch data was obtained. However, this means that process conditions are set according to production objectives, which in general means keeping conditions the same from batch to batch. This is a challenge for modelling the outcome of the process. This difficulty was tackled by using production data for several products to obtain a greater range of production conditions and response variables. However, it is likely that the models were still limited by the lack of variation between batches of the same product.

Ideally, data would be collected from a designed experiment were process conditions are varied to optimise discovery of causal factors. However, this would require great expense and resources, especially if all possible factors must be investigated. This project has lead to greater understanding about which factors in the process

impact DA and DE, and therefore provides a basis for future investigation.

CHAPTER 6

A new method for monitoring batch processes

In the previous three chapters, the results from research using real industrial bio-process data have been presented. The challenges encountered during the collaborations were discussed, and adapting to these challenges shaped the direction of research that was taken. The lack of available "bad" batches, and the amount of data available for specific products limited the research potential with regards to the classical monitoring methods discussed in Chapter 2. Monitoring and fault-detection of batch bio-processes have been of great academic interest, and as the real industry data was not suited to this, it was decided to pursue this research topic using simulated data. A valuable resource for simulated bio-process data was found to be the work of Van Impe and Gins (2015). Here, the authors provide a dataset of process data from simulated batch production of penicillin. This dataset includes batches from normal operating conditions, as well as batches displaying a wide variety of faults of different magnitudes and onset times. Using this dataset, a new method for monitoring batch processes was developed. The method was based on using the dynamic time warping distance as a similarity measure between batches, so that a fault is identified when the distance between an ongoing batch, and all previous successful batches becomes too great. The method was contrasted with multi-way PCA based monitoring, and validated using the faulty batches in the dataset. The journal article that resulted from this research is included in the following section.

6.1 Paper 3: Monitoring batch processes with dynamic time warping and k-nearest neighbours

Spooner, M., and M. Kulahci. 2018. "Monitoring batch processes with dynamic time warping and k-nearest neighbours". *Chemometrics and Intelligent Laboratory Systems* 183:102-112.
doi: https://doi.org/10.1016/j.chemolab.2018.10.011

Monitoring batch processes with dynamic time warping and k-nearest neighbours

Max Spooner[a,*], Murat Kulahci[a,b]

[a] *DTU Compute, Technical University of Denmark, Kgs. Lyngby, Denmark*
[b] *Department of Business Administration, Technology and Social Sciences, Luleå University of Technology, Luleå, Sweden*

Abstract

A novel data driven approach to batch process monitoring is presented, which combines the k-Nearest Neighbour rule with the dynamic time warping (DTW) distance. This online method (DTW-NN) calculates the DTW distance between an ongoing batch, and each batch in a reference database of batches produced under normal operating conditions (NOC). The sum of the k smallest DTW distances is monitored. If a fault occurs in the ongoing batch, then this distance increases and an alarm is generated. The monitoring statistic is easy to interpret, being a direct measure of similarity of the ongoing batch to its nearest NOC predecessors and the method makes no distributional assumptions regarding normal operating conditions. DTW-NN is applied to four extensive datasets from simulated batch production of penicillin, and tested on a wide variety of fault types, magnitudes and onset times. Performance of DTW-NN is contrasted with a benchmark multiway PCA approach, and DTW-NN is shown to perform particularly well when there is clustering of batches under NOC.

Keywords: batch process, dynamic time warping, nearest neighbours, pensim

1. Introduction

Batch processes are a widespread method of production in the manufacturing, food and medicine industries among others. In batch processes, quantities

*Corresponding author
Email address: mpsp@dtu.dk (Max Spooner)

of raw materials are subjected to a sequence of steps and conditions over a finite duration to transform them into the final product. Successful batches should meet product specifications and display as little variation as possible, and statistical process monitoring (SPM) is a useful tool for facilitating this goal. The data produced by a batch process has a three-way structure, consisting of I batches, for each of which J variables are measured at N time points throughout the batch's duration. Given this data, it is the goal of SPM to identify when a batch is out of control, .

Each mode of this data structure poses challenges to the application of SPM including high dimensionality in the variable mode (large J), with correlation between variables, different durations and rates of progress for different batches (varying N), the non-stationary and non-linear nature of the variable trajectories, changing covariance structure throughout a batch, and clustering of batches according to, for example, changes in raw materials, or maintenance schedules.

Nomikos and MacGregor [1] pioneered the application of SPM to batch processes with their development of the multiway principal component analysis (MPCA) monitoring scheme. This method deals with the three way data structure through unfolding such that the time dimension is combined with the variable dimension (often referred to as batch-wise unfolding). A PCA model is built on the unfolded data from a set of historical batches known to represent normal operating conditions (NOC). The model can then be used online during future batches, in order to detect deviations from NOC behaviour. Wold et al. [2] proposed a method of unfolding (dubbed variable-wise unfolding) whereby the batch dimension is combined with the time dimension prior to PCA model fitting and monitoring. However, this approach assumes, unrealistically, the same correlation structure between variables at every time-point in the process[3], and so the batch-wise unfolding approach[1] is adopted as the benchmark method in this work.

In [1], batches of equal duration were assumed and are required for the MPCA method. Subsequently, the issue of how to deal with variation in batch duration was widely investigated, leading to a number of different methods for

aligning batch process data. One approach made use of a so-called indicator variable, and re-sampled the data at even increments of this variable to obtain batches of the same length[4, 5]. In the chemometrics field, correlation optimised warping (COW) was developed for correcting peak-shifts in chromatographic data [6], and this method was later applied to aligning batch process data [7]. Another alignment method known as dynamic time warping (DTW) was applied to batch process SPM by Kassidas et al. [8]. This technique can synchronise trajectories of different lengths, and provides a distance measure quantifying the similarity of the trajectories. Online application of DTW is especially challenging as it entails aligning an incomplete batch to the most appropriate sub-section of a completed batch. Adaptations to the DTW algorithm have been proposed [9, 10] which aim to improve performance of online-alignment by limiting the degree of permitted re-alignment at each time-point. Other authors have proposed methods which avoid explicit synchronisation of the data, for example by applying feature extraction[11]. In this work, the distance measure provided by DTW is used to quantify an ongoing batch's distance to NOC batches so that the DTW alignment itself is the basis of the monitoring framework. As the effect of alternative DTW implementations such as [9, 10] on the DTW distance is uncertain, the classic DTW algorithm as originally introduced to batch process data[8] is used here.

DTW was developed in the field of speech recognition [12] in order to provide a distance measure to quantify the similarity of two time series trajectories that is not too sensitive to small timing differences. This enables the audio signal of a spoken word to be classified according to its similarity to the audio signals in a labelled reference database. It has since become a widespread technique in all fields related to clustering or classifying time series data [13]. As such, in most applications the synchronisation of the two trajectories is only a means to an end of obtaining the DTW distance. Conversely, in the batch process SPM field, it is the synchronisation step as a preprocessing tool, and not the DTW distance which has been the focus [8].

The nearest neighbour principle has been previously applied to batch process

data by He and Wang [14], who note that the approach is better suited to non-linear and multimodal data than MPCA. However, the Euclidean distance was used which has been shown to be highly sensitive to timing differences [15] and online implementation of the method during an ongoing batch was not discussed.

A novel method for online statistical process monitoring of batch processes using DTW and kNN is presented. The method, which we refer to as DTW-NN, compares an ongoing batch to all previous NOC batches using DTW. If the sum of the DTW distances to the k nearest NOC batches suddenly increases, the batch is signalled out of control. This data-driven approach is conceptually simple, as an ongoing batch is compared directly to previously completed NOC batches, and their similarity assessed via DTW. In addition to the DTW distance monitoring statistic, the identification of the most similar historical batches provides the basis for a useful visualisation tool for process operators. The DTW-NN method makes no assumptions regarding the structure of NOC batch data, and is robust to the presence of clustering and non-linearity provided that enough representative data is available. The method is highly flexible as the reference database can be continually updated with newly completed NOC batches.

In the following sections DTW-NN is outlined in detail. It is applied to the extensive datasets from [16] of a simulated penicillin production batch process and performance is contrasted with that of a benchmark MPCA scheme.

2. Methods

Given a set of class-labelled points, x_i for $i = 1, ..., n$, the k-Nearest Neighbour (kNN) classifier assigns a class to a new unlabelled point x_{n+1} according to the majority class of the k points closest in distance to x_{n+1} [17]. The Euclidean distance is often the distance measure used to determine the k-nearest neighbours. When each x_i is a time series, Wang et al. [15] have demonstrated the superiority of the DTW distance over the Euclidean distance for kNN classification for a wide range of applications and this distance measure will be discussed

in detail in the following section.

It is the goal of fault detection of batch processes, to assign to a batch the class labels of either "NOC" or "Faulty". However, the available training data usually only consists of NOC batches and few if any faulty batches. Even if some faulty batches are available, if the classical kNN classification method is used on a new batch exhibiting an unseen type of fault, it is unlikely to be effective. Therefore, it is preferable that the fault detection method is built using only NOC data, in order to detect any deviations from NOC behaviour. He and Wang [14] adapted the kNN principle for fault detection by using the sum of distances to the k nearest NOC batches. The distance between two batches was defined as the Euclidean distance between the unfolded batch data. The method was applied to complete batches of a semiconductor etch process. If the batch is faulty, this distance is expected to be large, where as if it is NOC there are likely to be similar past batches so the sum of distances will be small. The control limit for the sum of distances can be estimated from the distribution of its value for completed NOC batches.

We propose using DTW to identify the nearest neighbour NOC batches. DTW is well suited to the problem because batch processes often display variation in their rates of completion. A batch which is slightly slower than another batch but which otherwise has similar variable magnitudes and shapes would often result in a large Euclidean distance. In contrast, DTW is robust to small shifts, contractions and dilations in time and would therefore assign a small distance between the batches. Therefore DTW is a more appropriate measure of batch similarity. Furthermore, DTW can be used to estimate the similarity between a partially complete batch, and a complete batch whilst identifying the corresponding point in the complete batch which has been reached by the partial batch. This allows us to identify the nearest neighbour NOC batches, whilst a batch is in progress, in a way which is not sensitive to small differences in the speed of the ongoing batch. We focus on online monitoring which has greater relevance for those batch processes which take hours or days to complete and where there is time to take corrective action in response to any detected

faults.

Existing methods involving DTW and batch process monitoring, rely on a single reference batch to which all other batches are aligned. The reference batch should be a typical NOC batch. A problem with this is that many processes do not have stable conditions. For example there may be seasonal variation, sensor drift, wear on equipment or changes in suppliers of raw materials which result in sudden or gradual changes in normal operating conditions. Such variation can lead to multi-modal distribution in the database of previous batches. The reference batch which was a typical batch at one time, may later be very atypical.

In the kNN approach, instead of using a single reference batch, we compare an ongoing batch to all past NOC batches. The basic outline of DTW-NN has two steps

1. At current time t of an ongoing batch, perform open ended DTW to each past NOC batch, obtaining the DTW distance to each reference batch
2. Sum the k smallest DTW distances to obtain $D_t^{(k)}$. If $D_t^{(k)} - D_{t-1}^{(k)}$ exceeds the control limit, signal out of control

We found that it was better to monitor the rate of change in the sum of k smallest DTW distances, rather than the sum of distances directly.

In the following sections we outline the DTW algorithm, and procedures for selecting DTW parameter values. Then the DTW-NN model is presented with procedures for selecting the number of neighbours to use, k, and calculating the control limits.

2.1. Dynamic Time warping

After its development in the field of speech recognition [12], DTW has become widely used in batch process monitoring [8, 18, 9, 19, 20] to process the batches such that they have the same number of observations, and so that events within each batch are synchronised. This is done by selecting a single reference batch, and aligning the other batches to this reference batch. Once same-length batches are obtained, bilinear monitoring methods like MPCA can be applied.

Two multivariate sequences, $\mathbf{X}$ (m samples $\times$ J variables) and $\mathbf{Y}$ (n samples $\times$ J variables) can be synchronised using DTW as follows. First the $n \times m$ local distance matrix $\mathbf{C}$ is constructed, where $C_{i,j}$ is the distance between the i^{th} row of $\mathbf{X}$ and the j^{th} row of $\mathbf{Y}$ (denoted by $\mathbf{x}_i$ and $\mathbf{y}_j$ respectively). For batch process data, usually the weighted squared Euclidean distance is used

$$C_{i,j} = (\mathbf{x}_i - \mathbf{y}_j)^T \mathbf{W} (\mathbf{x}_i - \mathbf{y}_j) \tag{1}$$

where $\mathbf{W}$ is a $J \times J$ diagonal matrix containing the variable weights, to be discussed in the following section.

DTW considers warping paths, f, through the local distance matrix $\mathbf{C}$ that represent mappings, between $\mathbf{X}$ and $\mathbf{Y}$: $f = f_1, f_2, \ldots, f_T$ where $f_t = (i,j)_t$. The warping path is usually subject to the following constraints:

- Boundary conditions: $f_1 = (1,1)$ and $f_T = (m,n)$, i.e., the first elements of $\mathbf{X}$ and $\mathbf{Y}$ should be aligned to each other as well as their last elements
- Continuity: If $f_t = (i,j)$ then $f_{t+1} = (i+a, j+b)$ where $a, b \leq 1$, i.e., the warping path cannot skip cells in the local distance matrix
- Monotonicity: If $f_t = (i,j)$ then $f_{t+1} = (i+a, j+b)$ where $a, b \geq 0$, i.e., each pairing cannot go backwards in time in either $\mathbf{X}$ or $\mathbf{Y}$.

The accumulated distance between the two sequences under a warping, f, is given by

$$D_f(\mathbf{X}, \mathbf{Y}) = \frac{1}{n+m} \sum_{t=1}^{T} \omega_t C_{f_t} \tag{2}$$

where $\omega_t = 2$ if warping the function from f_{t-1} to f_t takes a diagonal step and $\omega_t = 1$ otherwise. These step weights are used to compensate for warping paths of different lengths [12]. DTW identifies the warping function which minimises the accumulated distance D_f and the corresponding value

$$D = \min_f [D_f(\mathbf{X}, \mathbf{Y})] \tag{3}$$

is the DTW distance between the two sequences. The optimal warping function is found using dynamic programming [12], so only a small subset of all possible f need be considered in order to solve Eq. 3.

Additional constraints can be imposed on DTW to limit the amount of warping in the solution. For example, global constraints can be used to limit the maximum absolute time difference between the warped sequences and local constraints limit the slope of the warping function such that extreme contractions or expansions are not permitted.

To apply DTW online, whilst a batch is in progress, open-ended DTW can be used [8]. In this version of DTW the end-point boundary condition is relaxed so that $\mathbf{X}$ is aligned to a leading portion of $\mathbf{Y}$. Let $\mathbf{Y}^{(p)}$ $(p \times J)$ denote the first p samples of $\mathbf{Y}$, then open ended DTW solves

$$D = \min_{f,p}[D_f(\mathbf{X}, \mathbf{Y}^{(p)})] \tag{4}$$

The resulting value of p is the time-point in the reference batch which best corresponds to the current time in the ongoing batch.

2.2. DTW Parameters

The multivariate form of DTW should be used as for most batch processes several variables are present, and it is the aim that the DTW distance will be sensitive to disturbances in any variable. The variables may be scaled to remove the extreme differences in magnitude due to different engineering units. An appropriate scaling is to divide each variable by its mean range (found from calculating the range of each variable for each batch in the reference dataset, and taking the mean of these ranges)[8]. Note that DTW-NN is independent of the variable scaling used as the scaling effect is taken into account by the DTW variable weights, $\mathbf{W}$ in Eq. 1. Various approaches to calculating $\mathbf{W}$ were investigating whilst developing the DTW-NN method. Previous weighting methods [8, 18], were developed in the context of using DTW for data alignment, and so aim to up-weight variables which are better guides for alignment, and down-weight those which are not. However, applying these weighting methods in the DTW-NN monitoring approach lead to poorer performance, as they result in the DTW distance being dominated by only a few of the variables, hindering the detection of faults which do not affect these specific variables.

Instead, a new weighting method which ensured that all of the variables contribute approximately equally to the DTW distance was found to lead to better performance.

To encourage a more equal contribution to the DTW distance, the variables that vary most from batch to batch should have a small weight in the local distance calculation of Eq. 1, and variables that have least batch to batch variation should have greater weight. In batch processes, the batch to batch variation of each variable may be dependent on time. Let $\sigma^2_{j,t}$ denote the variance across the I reference batches of variable j at time t. Then let

$$\tilde{\sigma}^2_j = \text{median}(\sigma^2_{j,t}) \tag{5}$$

denote the median of the batch-to-batch variance of variable j over the entire batch duration. The reason for taking the median batch-to-batch variance over the process duration, is to reduce the influence of extreme values, occurring for example in periods containing phase transitions. The median provides a more robust estimate of the average batch-to-batch variance over the entire process duration. Then we define $\mathbf{W}$ as the diagonal matrix where

$$W_{j,j} = \frac{1}{\tilde{\sigma}^2_j} \tag{6}$$

In addition, if batches appear to cluster into groups, the weights should be adjusted by calculating $\tilde{\sigma}^2_j$ for each cluster in the data and then averaging across clusters. This ensures that the weights are not distorted by between-cluster variation. When there is no prior knowledge regarding the presence of clusters, and which batches belong to which clusters, hierarchical clustering can be performed on the reference dataset, with the DTW distance in Eq. 3 as the between-batch distance measure. The DTW distance should be calculated using a subset of the J variables which are most likely to be affected by clustering. The silhouette method from [21] may be used to judge whether clustering is present, and if so how many clusters. If clusters are detected, then $\tilde{\sigma}^2_j$ from Eq. 5 is calculated for each cluster, using only the batches in that cluster, and then the average of $\tilde{\sigma}^2_j$ is taken across clusters before applying Eq. 6. For further

details regarding hierarchical clustering and the silhouette method we refer to [21].

A local constraint should be used in order to avoid singularities in the DTW alignments. Without a local constraint, the alignment is free to compress long sections of the batch to an instant, or "freeze" one time sample for a prolonged duration, in order to synchronise two batches. It is expected that batches run faster or slower than each other, but not that they stop completely or suddenly jump forward in time. The local constraints suggested in [12] are parametrised by the value P, which represents the number of diagonal steps the warping function must take for each horizontal or vertical step. A method for selecting the value of P may be found in [20] and is applied in this work.

Finally, a global constraint may be applied to shorten computational time as it reduces the need to calculate every pairwise local distance. We use the simplest type of global constraint known as a band constraint. This limits alignment of point t in an ongoing batch to some point in the interval $[t-w, t+w]$ in the reference dataset batch. For w we used the range of the durations of the reference batches.

2.3. DTW-NN model specification

Suppose a dataset of I NOC batches is available. Then, the DTW-NN model is specified as follows.

First, the I batches are used to determine the DTW parameters ($\mathbf{W}$, local constraint and global constraint) as described in the previous section. Then the first batch is treated as an ongoing batch $\mathbf{X}$, and the remaining $I-1$ batches as the reference dataset. For each sample time t in $\mathbf{X}$, open ended DTW is performed between the first t rows of $\mathbf{X}$ and every other batch in the reference dataset and the resulting DTW distances are saved. At each time point the smallest k distances are summed to obtain the quantity $D_t^{(k)}$, which is a measure of the similarity between the entire variable trajectories of the ongoing batch up to time t, and k nearest NOC batches. If a fault occurs at time $t+1$ which influences one or more of the variable trajectories, $D_{t+1}^{(k)}$ may not

necessarily immediately jump to a large value, because it takes into account the similarity over the entire batch duration so far, which up until time $t+1$ was entirely normal. However, we expect to observe an increase in $D_{t+1}^{(k)}$ relative to its previous value. Therefore, the change in nearest neighbour distance is calculated as

$$\dot{D}_1^{(k)} = D_1^{(k)} \tag{7}$$

$$\dot{D}_t^{(k)} = D_t^{(k)} - D_{t-1}^{(k)} \text{ for } t = 2, \ldots, N \tag{8}$$

At the end of the batch, let $D^{(k)}$ and $\dot{D}^{(k)}$ be vectors where the t^{th} elements are $D_t^{(k)}$ and $\dot{D}_t^{(k)}$ respectively. This procedure is repeated for the remaining batches, such that each in turn is treated as the ongoing batch and $D^{(k)}$ and $\dot{D}^{(k)}$ vectors are obtained for each batch in the reference database.

The $\dot{D}^{(k)}$ vectors are arranged in a $I \times N$ matrix, $\dot{\mathbf{D}}$, where the i^{th} row of $\dot{\mathbf{D}}$ is $\dot{D}^{(k)}$ for the i^{th} batch, and N is the length of the longest reference batch. The duration of batches can vary so there may be missing values in the final columns of $\dot{\mathbf{D}}$. As $\dot{\mathbf{D}}$ contains the rates of change of the kNN distance throughout each NOC batch in the reference dataset, it can be used to calculate $100(1-\alpha)\%$ control limits. The rate of change in kNN distance is time dependent and is typically larger at the start of a batch. Therefore, time dependent control limits are calculated. The theoretical distribution $\dot{D}_t$ is not known so the quantile is calculated empirically for each column of $\dot{\mathbf{D}}$. A Gaussian kernel is used to smooth the empirical probability density of each column and obtain more reliable estimates of the extreme quantiles used for control limits (e.g. $\alpha = 0.01$). For the final columns of $\dot{\mathbf{D}}$ that contain missing values, the control limits can still be calculated using only the non-missing values. The accuracy of the control limits will decrease as the number of missing values increases, but so too will the expected number of future batches that will still be in progress for those time-points.

The number of nearest neighbours, k, influences the smoothness of $D^{(k)}$ so to increase stability k should be greater than one. Conversely, if k is too large, $D^{(k)}$ becomes less sensitive to local structures in the reference dataset and also

k needs to be less than the size of any known batch clusters. We devised the following unsupervised method for selecting k.

Consider the first nearest neighbour of a batch in the reference dataset. The first nearest neighbour is likely to change throughout the batch duration. Let κ_i represent the number of unique batches observed within the first nearest neighbour to batch i throughout the duration of batch i. Calculate κ_i for $i = 1, \ldots, I$. Let $k = \text{mode}(\kappa_i)$.

The motivation for this approach is that we often observed the nearest neighbours to change places with each other over the course of a batch. For example, for time 1 to 50 batch 11 may be the first nearest neighbour and batch 32 the second nearest neighbour, then at time 51, batch 32 is the first nearest neighbour and batch 11 the second nearest neighbour. By letting $k = \text{mode}(\kappa_i)$ it is hypothesised that for the most part the k nearest neighbours will consist of the same k batches throughout the duration of the batch, and this will minimise instability caused by changes in members of the set of k nearest neighbours. It should be noted that the value of k obtained using the above method will generally increase with I, the number of batches in the reference set. This is reasonable, because a bigger reference dataset will contain a greater number of batches that are similar to a new batch, so the monitoring statistic should take into account the distance to more nearest neighbours.

2.4. Monitoring a new batch with DTW-NN

To monitor a new batch using DTW-NN, let $\mathbf{X}$ be a new batch for which data has been collected up until the t^{th} sample. Then the DTW distance is calculated between $\mathbf{X}$ and each batch in the reference dataset. The smallest k distances are summed to obtain $D_t^{(k)}$ and the change in the nearest neighbour distance is calculated using (7) to obtain $\dot{D}_t^{(k)}$. Finally, an out of control signal is generated if $\dot{D}_t^{(k)}$ exceeds the control limit for time t.

In addition to the kNN distance statistic, this model provides valuable information for online data visualisation by identifying the k nearest neighbours at each time-point of an ongoing batch. The variable trajectories of the k nearest

neighbour batches can be plotted together with the partially complete trajectories of the ongoing batch to provide insight to operators on how the ongoing batch compares to historical data.

2.5. MPCA

To assess the performance of the proposed method, we compare it to the MPCA method introduced by [1, 22], where the data is aligned with DTW following [8]. This method is based on fitting a PCA model to the unfolded NOC reference dataset as follows.

First the I reference batches are scaled by dividing each variable by its mean range over the I batches. Next, a reference batch is selected, to which the remaining batches are aligned using DTW such that all batches then have the same number of observations as the reference batch, N. The variable weights, $\mathbf{W}$ used in this alignment are here determined using the method in [18] which was developed to give greater weights to those variables which are most indicative of process time. To avoid extreme warpings, a local constraint is selected according to [20], and a band global constraint of width equal to the range of the batch durations is used.

Let $\underline{\mathbf{Z}}$ $(I \times J \times N)$ be a three-way data matrix formed by the aligned NOC reference data set (where N is the length of the reference batch). This matrix is unfolded batch-wise to form $\mathbf{Z}$ $(I \times JN)$. The columns of $\mathbf{Z}$ are mean centred and scaled to unit variance and a PCA model is fitted, where L components are retained. The loading vectors from this model, as well as the column means and standard deviations, are saved for the online implementation.

For online monitoring of an ongoing batch at time t, the first step is to align the existing section of ongoing batch to the reference batch using open-ended DTW. The aligned data is then unfolded, centered and scaled using the previously saved parameters. As the ongoing batch is only partially complete, it is necessary to estimate the future observations in some way so the the scores can be calculated using the PCA model. We use the projection to the model plane approach from [22]. Then the PCA model is applied to obtain the current scores

and the monitoring statistics of Hotelling's T^2 and Q (the squared prediction error of the model at the current time point). Following [23, 24], we used a time dependent mean and covariance to calculate T^2 which is determined from the reference dataset using a leave one out procedure. The $100(1-\alpha)\%$ control limit for Hotelling's T^2 is given by

$$T^2_\alpha = \frac{L(I^2-1)}{I(I-L)} F_{L,I-L}(\alpha) \tag{9}$$

where $F_{L,I-L}(\alpha)$ is the $(1-\alpha)$ quantile of the F distribution with L and $I-L$ degrees of freedom.

The control limits for Q were calculated using the standard approach based on the $g\chi^2_h$ distribution with g and h estimated for each time point from the reference dataset using the method of moments [22].

3. Results and Discussion

3.1. The Pensim datasets

To evaluate the proposed method, the four simulated datasets provided by [16] were used. These datasets, which extend the Pensim model of [25], represent data from batch production of penicillin. The simulation was designed to be as realistic as possible, for example, with regards to sensor measurement noise, batch to batch variability of initial conditions and automatic control mechanisms. Each of the four datasets represents different assumptions about the production process which are briefly summarised as follows

- Dataset 1: "Base". Initial conditions independently sampled from normal distributions
- Dataset 2: "Skewed". Some of the initial conditions sampled from uniform distribution, and non-linear dependence introduced between some initial conditions
- Dataset 3: "Tail". One of the parameters of the Pensim model is drawn from a χ^2 distribution. Initial conditions are generated the same way as for dataset 2.

- Dataset 4: One of two types of micro-organisms is used for each batch. Initial conditions are generated the same way as for dataset 2.

Each dataset consists of 400 NOC batches and several thousand faulty batches. The faulty batches consist of 15 different fault types, each with a range of different fault magnitudes. Additionally, the faults start in one of four different time intervals. The original dataset contains 50 repetitions of each type/magnitude/onset-time-interval combination for a total of 22,200 faulty batches. To save computational time, we reduced this to 5 repetitions per combination by random sampling, resulting in a total of 2220 faulty batches per dataset. For further details on the simulation of the datasets we refer to the original work [16].

Table 1: Variables measured online in Pensim process and abbreviations used in this paper

Variable	Abbreviation
Time (h)	Time
Volume (m^3)	Volume
Dissolved Oxygen concentration (mg/L)	O2
Dissolved CO_2 concentration (mg/L)	CO2
Reactor temperature (K)	RctrTemp
pH (-)	pH
Feed rate (L/h)	FeedRate
Feed temperature (K)	FeedTemp
Agitator power (W)	Agitator
Cooling water flow rate (L/h)	WaterRate
Base Flow Rate (L/h)	BaseFlow
Cumulative base flow (mL)	BaseQty
Cumulative acid flow (mL)	AcidQty

The 12 variables listed in table 1 are measured online throughout each batch and will be used for monitoring. In the original dataset the variables are measured every 0.2 hours resulting in around 2300 samples per batch. To reduce

noise, the sampling frequency was reduced to hourly by taking the mean every 5 samples. After this, the number of samples (now equivalent to the duration in hours) of the 400 NOC batches ranged from 454 to 467.

Of the 400 NOC batches, 50 were randomly selected to use as the reference dataset. As the magnitudes of the 12 different variables varied greatly, they were scaled by mean variable range of the 50 batches in the reference dataset.

In the following sections we present detailed results of applying DTW-NN and MPCA to dataset 4 which displays the multi-modal features that we expect DTW-NN to be especially suited to. In section 3.6 we summarise the findings of applying the methods to all four datasets.

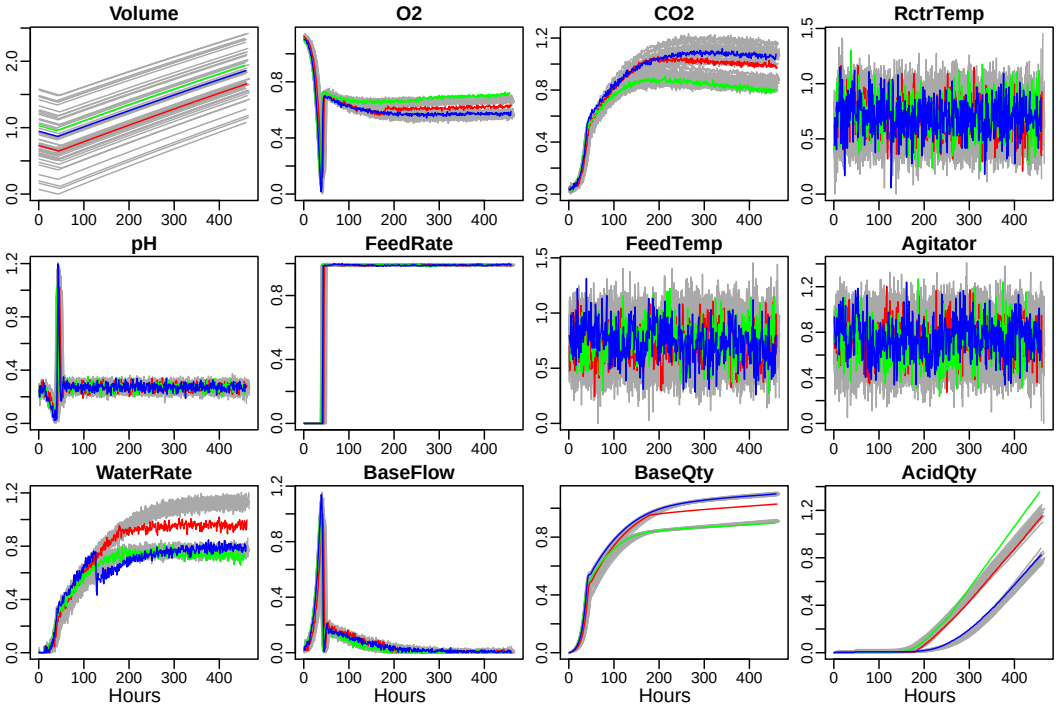

Figure 1: Variable trajectories of the 50 NOC batches randomly selected as the reference dataset (grey), and 3 faulty batches (red, blue, green), after scaling by mean range.

The 50 reference batches for dataset 4 are plotted in Fig. 1. Some of the

variables (O2, CO2, WaterRate, BaseQty, AcidQty) show multi-modality as a result of the two different strains used for different batches. In a real scenario, this type of variation may be observed due to using different suppliers, or different batches of raw material from the same supplier. Monitoring performance will be summarised for all 2220 faulty batches, but three faulty batches were selected for more detailed illustration and are also shown in Fig. 1. These are the red batch (fault type 1: -10% change in feed substrate concentration at time 178 hrs), the blue batch (type 2: -10% in coolant temperature at time 125 hrs) and the green batch (type 15: simulated contamination at time 254 hrs).

3.2. DTW parameter tuning

As there is clearly clustering in the data, it was appropriate to apply hierarchical clustering before calculating the variable weights. The dataset contained information on which strain was used for each batch, so the correct clustering is known, but we chose to try and identify the clustering empirically without using this information. To perform the clustering, the DTW distance was calculated between every pair of batches in the reference database, based on the variables BaseQty, WaterRate and O2. The silhouette method clearly identified two clusters as most appropriate, so the hierarchy was cut at two clusters. Comparing to the strain labels showed that this approach had correctly classified all the batches. It must be noted that in this dataset the clustering of batches, and the variables most indicative of clustering, are quite easily visible in Fig. 1. Other cases may pose a greater challenge to successful clustering. In such cases, prior knowledge of possible clusters due to, for example, different suppliers of raw materials, would be useful.

Next, for each variable, and for each cluster, the median over time of batch to batch variance was calculated (Eq. 5). Then, the average of the variance estimates from the two clusters was taken before calculating the final weights (Eq. 6). The resulting weights are shown in Fig. 2.

Next, using the method in [20], a local constraint of $P = 1$ was selected. This constraint enforces one diagonal step for each vertical or horizontal step

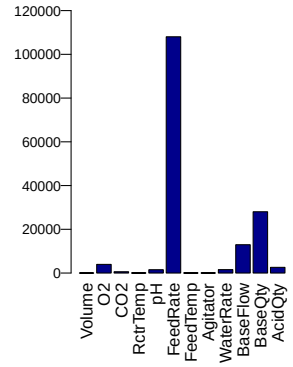

Figure 2: Variable weights for DTW

taken by the warping function.

Finally, the range of the durations of the 50 reference batches was 12. Hence the shortest batch was 12 hours shorter than the longest batch. If a new batch is in NOC we will not expect it to be more than 12 hours faster or slower than any of the reference batches at any point, and therefore a band global constraint of ± 12 samples was selected for DTW.

3.3. DTW-NN model specification

For each batch in the reference database, online DTW was performed to the other 49 reference batches, and the nearest neighbours to the batch were identified throughout its duration. To select k, the number of unique batches identified as the first nearest neighbour was examined. For example, Fig. 3 shows the online DTW distance throughout batch 19 to every other reference batch. The nearest neighbour batch to batch 19 changes several times but from $t = 70$ onwards remains constant at reference batch 28. In total, 5 different reference batches were identified as the nearest neighbour to batch 19 throughout its duration. The distribution of the number of unique batches appearing as the nearest neighbour throughout every reference batch is shown in Fig. 4, and we selected $k = 8$ from the mode of this distribution.

Next, the sum of the 8 nearest neighbour distances, $D^{(8)}$ was calculated

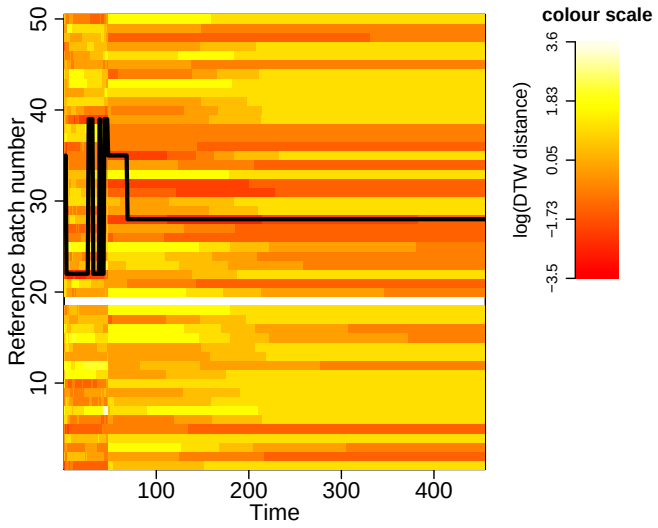

Figure 3: Heatmap of the online DTW distance between batch 19 and the other 49 batches in the reference dataset, throughout the duration of batch 19. The black line indicates the nearest neighbour to batch 19 at each time point.

throughout each batch. The value is shown for the 50 reference batches in Fig. 5, as well as for the three faulty batches. This illustrates the motivation for monitoring the change in $D^{(8)}$ rather than $D^{(8)}$ itself, as after each fault a clear increase in $D^{(8)}$ is observed even though the value of $D^{(8)}$ is not necessarily large compared to the values observed for the NOC reference batches. Therefore, $\dot{D}^{(8)}$ was calculated. In order to compare performance over the same time periods as MPCA, we truncated $\dot{D}^{(8)}$ to the length of the shortest batch (455). To aid visualisation, $\dot{D}^{(8)}$ was mean centered and scaled to unit variance at each time point using sample statistics from the 50 reference batches. An $\alpha = 0.01$ control limit was calculated for each time point by smoothing the sample probability distribution of $\dot{D}^{(8)}$ from the 50 reference batches with a Gaussian kernel. This confidence limit, and $\dot{D}^{(8)}$ for the 50 reference and red, blue and green batches is shown in Fig. 5. Before evaluating performance across all 2220 faulty batches, results regarding specification of the MPCA benchmark model will be presented.

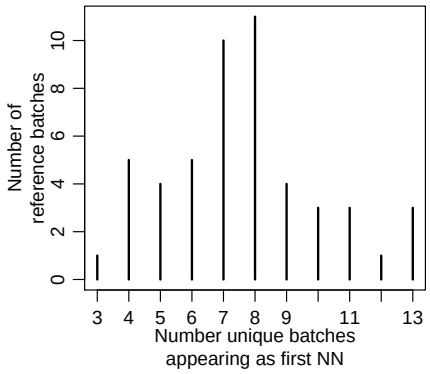

Figure 4: Distribution of the number of unique batches appearing as the first nearest neighbour of each reference batch

3.4. MPCA model specification

The MPCA monitoring model was built as described in section 2.5. For DTW alignment of the data, the reference batch was selected as the batch closest to median duration, and a local constraint of $P = 1$ was used following [20] and variable weights were determined using [18]. The reference dataset of 50 batches was aligned and unfolded such that each variable at reference times 1 to 455 was placed side by side. This 50×5460 matrix was centered and scaled and a PCA model was fit. Based on the scree plot of the eigenvalues (Fig. 6), 5 principal components were retained in the model. The scores were predicted online throughout all batches, with projection to the model plane as the infilling method. This included online alignment to the reference batch using open-ended DTW. Online Hotelling's T^2 and Q statistics were calculated from the scores, as well as their $\alpha = 0.01$ control limits.

3.5. Performance evaluation

Three key performance indicators (KPI) were adopted for the evaluation of monitoring performance:

- FAR (False Alarm Rate) = proportion of samples exceeding the control limits whilst the batch is in a NOC state.

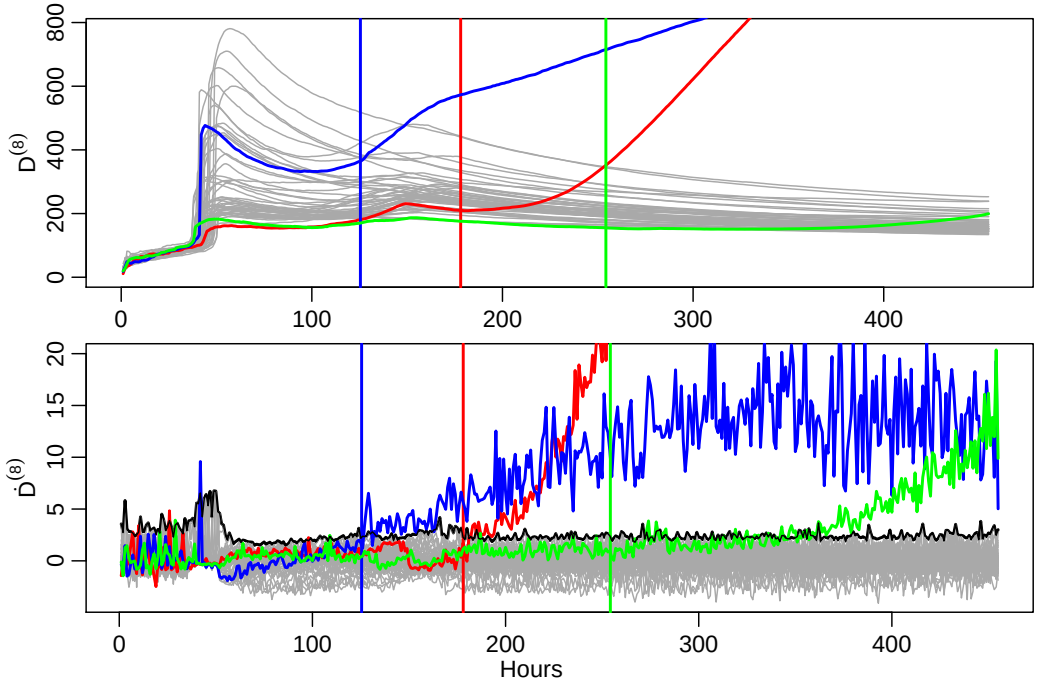

Figure 5: Sum of 8 nearest neighbour distances over time (top) and the slope over time (bottom) for the 50 reference batches (grey) and the red, blue and green faulty batches. The vertical lines indicate the fault onset times. The black line is the control limit.

- TDR (True Detection Rate) = proportion of samples exceeding the control limits whilst a batch is in a faulty state.
- TTS_a (Time To Signal) = the number of samples observed before a consecutive samples exceeded the control limits, since the onset of a fault. If a consecutive samples never exceed the limits we define TTS for that batch as ∞.

These indicators were calculated for each batch individually. The batches were defined as in a faulty state for all observations after the fault onset time, and as in a NOC state for all observations prior to the fault onset. The 400 NOC batches were in a NOC state for their entirety.

Table 2 contains KPI summary statistics for the DTW-NN and MPCA meth-

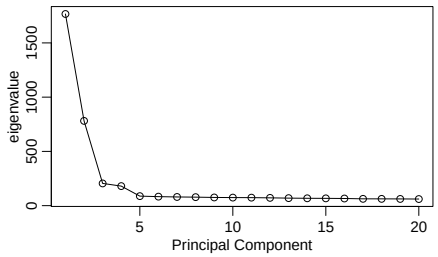

Figure 6: Scree plot of first 20 eigenvalues of the MPCA model

Table 2: Mean FAR of the 350 NOC batches not in the reference dataset, and mean TDR, median TTS_1, median TTS_3 of the 2220 faulty batches

	FAR	TDR	TTS_1	TTS_3
DTW_NN	0.0119	0.5457	8.0	32.0
MPCA_Q	0.0071	0.3403	16.0	>362.0
MPCA_T^2	0.0009	0.1613	>445.0	>344.0
MPCA_Q-or-T^2	0.0079	0.3517	16.0	>362.0

ods. In evaluating the two methods we must account for the fact that the MPCA model consists of two monitoring statistics, whilst the DTW-NN model has a single monitoring statistic. Table 2 shows that the mean TDR for the combined Q-OR-T^2 alarm is only marginally better than TDR for Q alone. In addition, in 2164 of the 2220 faulty batches, TDR for Q exceeded TDR for T^2. Therefore, the Hotelling's T^2 of the MPCA model is made almost entirely redundant by the Q statistic. For these reasons, we hereafter focus on comparing the performance of DTW-NN to the MPCA Q chart.

To make a fair comparison between two monitoring statistics, it is important that the FAR is similar for both of them. Table, 2 shows that mean FAR of DTW-NN is close to that of MPCA-Q, and they are both close to the target value of $\alpha = 0.01$. Their distributions also do not differ greatly (Fig. 7). Despite this, mean TDR is much higher for DTW-NN than for MPCA-Q, so we conclude that DTW-NN is the superior method for this dataset. A mean TDR of 54 % for DTW-NN may not appear large, but it is important to note that the dataset

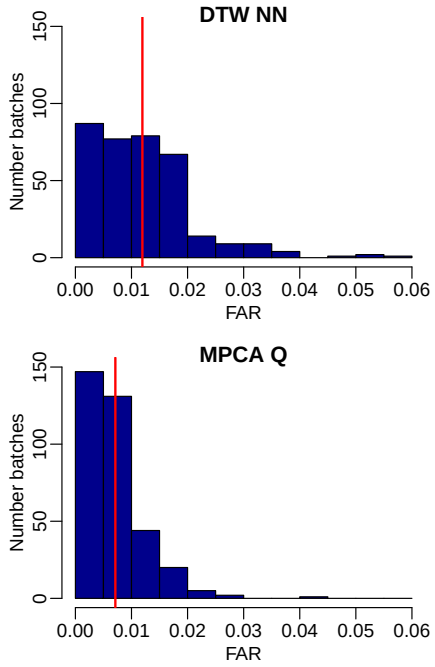

Figure 7: Distribution of FAR for the two methods, with mean value shown by the vertical lines

contains a range of fault magnitudes, many of which are very mild.

In Fig. 8 TDR of the two methods is plotted for all 2220 faulty batches, and shows that DTW-NN performance is as good as or better than MPCA-Q for 13 of the 15 fault types. Only for fault type 3 and fault type 9 (Agitator power drop and Non-functioning pH control respectively) does MPCA-Q appear to perform systematically better than DTW-NN. Fig. 9 shows the online monitoring statistics for the three example batches. For these batches, DTW-NN clearly signals a deviation from the behaviour of historical batches, whilst MPCA-Q does not detect anything unusual.

[1]The MPCA-Q TDR plot for Fault 6 reproduces closely the results presented in Fig. 14(d) of [16] with minor differences likely due to choice of MPCA infilling method, control limits, and our use of 5 batch repetitions rather than 50.

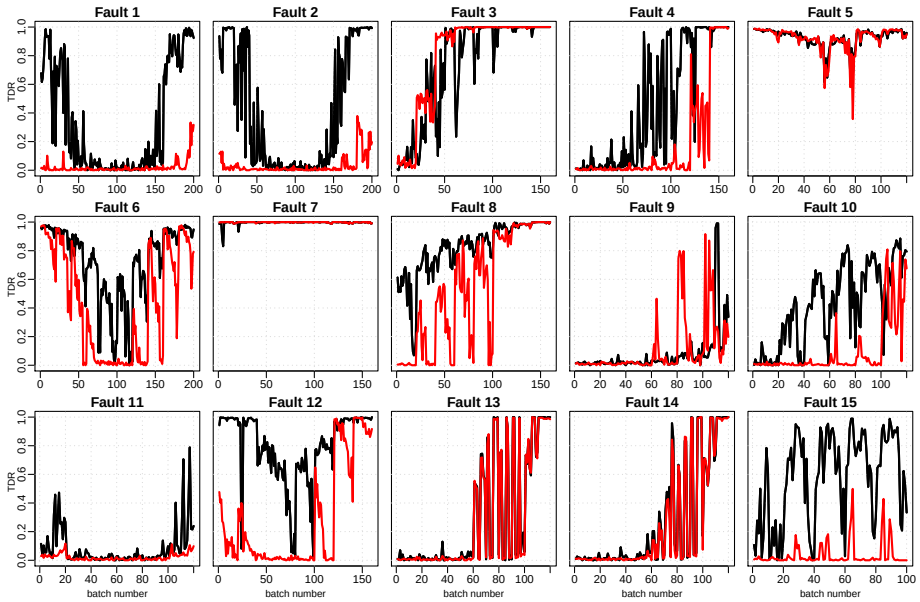

Figure 8: TDR of DTW-NN (black) and MPCA-Q (red) for all faulty batches[1]

The speed of detection for DTW-NN is also faster than for MCPA-Q with median TTS_1 values of 8 and 16 respectively (Table 2). For TTS_3, DTW-NN has a median value of 16 whilst most the median for MPCA-Q is over 344.

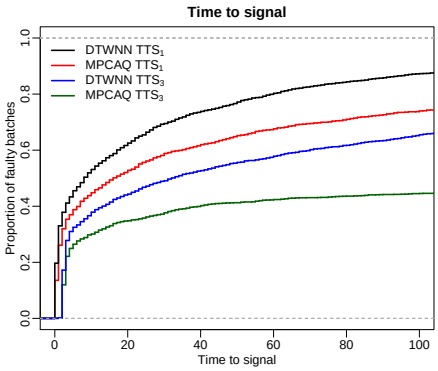

Figure 10: Observed cumulative distribution of TTS_1 and TTS_3

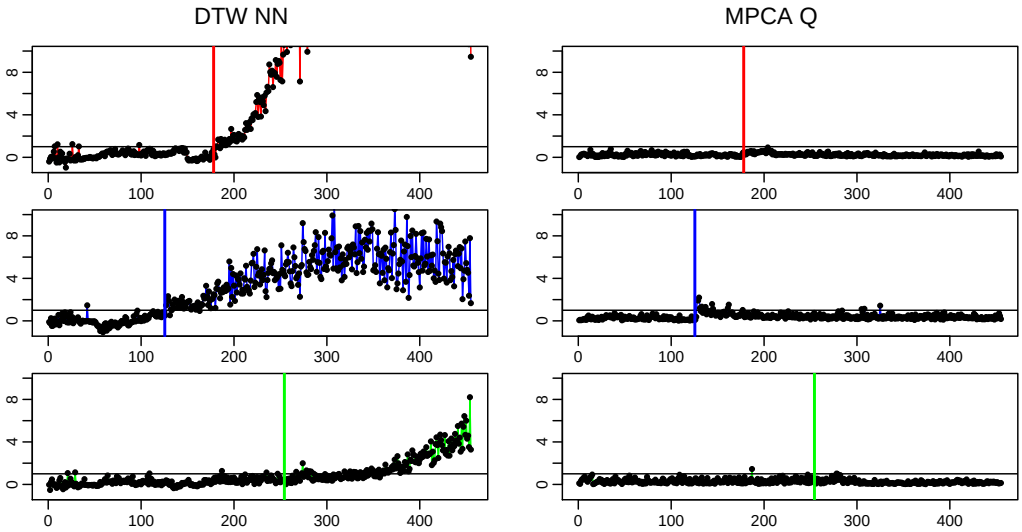

Figure 9: Monitoring results for the red (top), blue (center) and green (bottom) faulty batches with DTW-NN (left) and MPCA-Q (right). For easier comparison, the statistics have been divided by their control limit at each time point

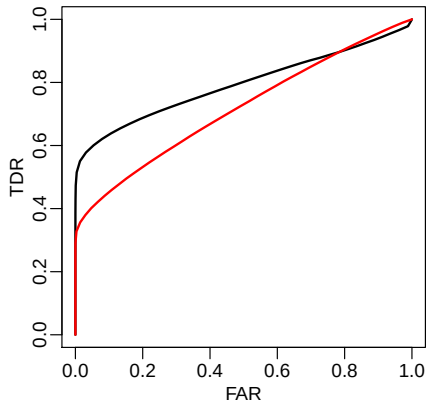

Figure 11: ROC Curve for DTW-NN (black) and MPCA-Q (red)

As discussed in, for example, [23], a useful tool for comparing monitoring methods independently of a specific FDR is the ROC curve. We constructed

the ROC curves in Fig. 11 by calculating control limits for many values of α in the range $[0, 1]$ and for each value calculating the mean TDR and FAR over all faulty batches. The ROC curve confirms that the better performance of DTW-NN on this dataset cannot be attributed to chance differences in control limits.

It was of interest to examine the effect of number of neighbours, k, on performance and check the suitability of the proposed method for selecting k from section 2.3. The preceding analysis was repeated for different k and the results are shown in Fig. 12. In fact, the selected value of $k = 8$ had the second-highest TDR value, whilst a value of $k = 7$ results in the highest TDR, though the difference in TDR is marginal (less than 0.001), Fig. 12 shows that TDR is not very sensitive to choice of k in the range $k = 2$ to 10. TDR for $k = 1$ is noticeably lower than $k = 2$. This may be because when a fault occurs, there is greater likelihood of finding a single past NOC batch that somewhat resembles the faulty batch, than there is of finding two NOC batches that resemble the faulty batch. Mean TDR drops to a much lower value for $k > 23$, as then $\mathbf{D}^{(k)}$ is forced to include distances to NOC batches in a different cluster than the ongoing batch.

The effect of the size of the reference dataset, I on the selected value of k using the proposed decision rule was examined by supposing the reference dataset consisted of $I = 1, 2, \ldots, 50$ reference batches and applying the rule in each case (Fig. 13). It appears that the value of k selected using the decision rule can be approximated by $\sqrt{I}$, and this provides a quicker method for specifying k, which would be useful if new batches are continuously added to the reference dataset in a production setting.

3.6. Other Pensim datasets

The results of applying the methods to datasets 1, 2 and 3 will now be presented together with the dataset 4 results. The same procedures were followed as previously described for dataset 4. As there is no indication of clustering of batches in datasets 1 to 3, the DTW weights were calculated using all 50 refer-

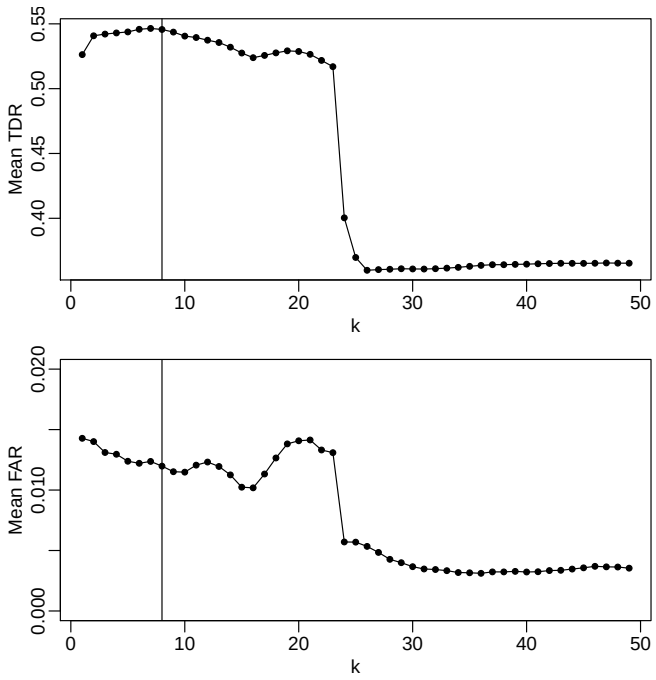

Figure 12: Mean TDR (top) and mean FAR (bottom) for $k = 1, \ldots, 49$. The vertical line shows the value used, $k = 8$.

ence batches, without the cluster identification step. The DTW-NN parameter k was selected in the same way for each dataset. The number of components retained in the MPCA model was 5 for each dataset, and the MPCA model was trained and applied as described previously.

The performance of both methods' monitoring statistics for all four datasets is summarised in Fig. 14 and we highlight the following observations

- MPCA-Q shows marginally higher TDR than DTW-NN on dataset 3, but substantially lower TDR on dataset 4
- DTW-NN shows a more stable TDR across the four datasets (ranging from 0.51 to 0.61) than MPCA-Q (ranging from 0.34 to 0.58)
- For all the datasets, monitoring both Q and T^2 from MPCA, results in only a minor improvement in TDR compared to only monitoring Q, and a substantially higher FAR (for datasets 1 to 3 in particular).

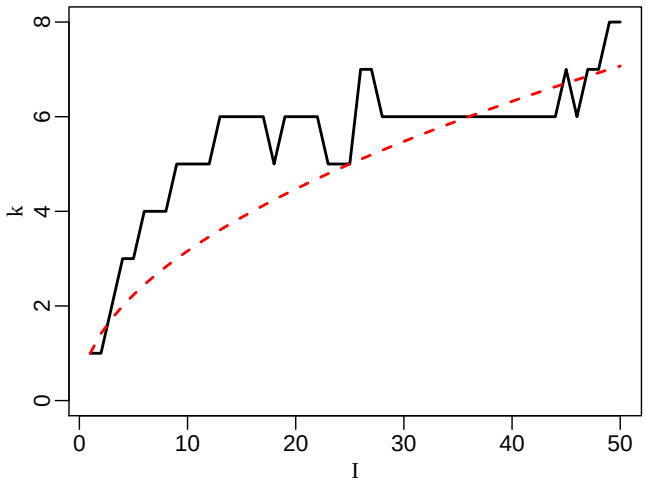

Figure 13: The value of k (solid black line) selected for difference sizes of reference dataset, I. For comparison $\sqrt{I}$ (dashed red line) is also shown

- For both MPCA-Q and DTW-NN, the observed mean FAR deviates from the target value of $\alpha = 0.01$ for different datasets. Mean FAR varies from 0.0071 to 0.0191 for MPCA-Q and from 0.0045 to 0.0153 for DTW-NN.

The validity of the proposed rule for selecting k was also examined for datasets 1 to 3, by comparing the selected value (k_{selected}) to the value which resulted in the greatest mean TDR (k_{maxTDR}) as shown in Table 3. For datasets 1 to 3, mean TDR for k_{selected} and k_{maxTDR} differs by at most 0.0144 (for dataset 3), supporting the view that mean TDR is not highly sensitive to k, and that the selection rule is a reasonable approach for selecting k that gives a close to optimal TDR.

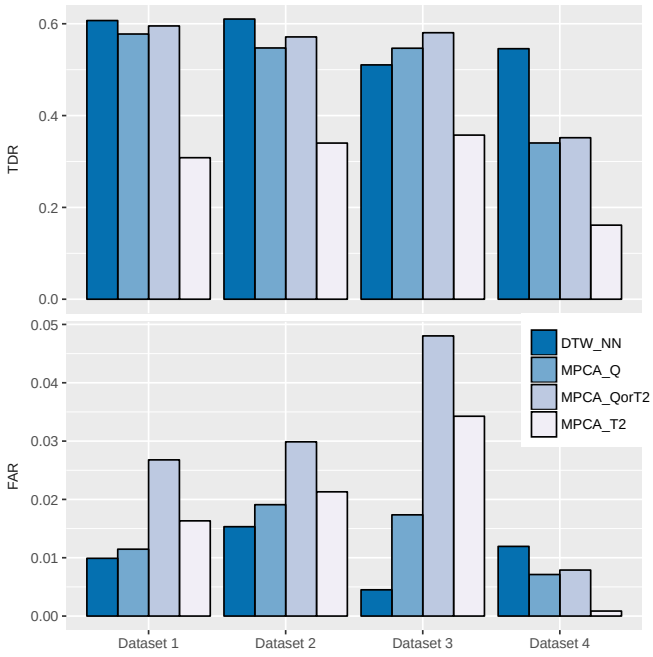

Figure 14: Mean TDR (top) and FAR (bottom) for MPCA and DTW-NN on all four sets.

Table 3: Selected k values, and maximum TDR k values for each dataset

Dataset	k_{selected} (mean TDR)	k_{maxTDR} (mean TDR)
1	8 (0.6070)	7 (0.6104)
2	6 (0.6102)	11 (0.6133)
3	6 (0.5103)	2 (0.5248)
4	8 (0.5457)	7 (0.5465)

3.7. Computational Time

DTW-NN is computationally intensive as the DTW distance must be calculated between the ongoing batch, $\mathbf{X}$, and each batch in the reference database $\mathbf{Y}_i$ repeatedly for each new sample in the ongoing batch. However, in an online implementation it is only necessary to add a new row to the local distance matrices $\mathbf{C}$ and accumulated distance matrices as each new sample of $\mathbf{X}$ is obtained,

rather than re-calculate the entire local distance matrices. The analysis was performed using the statistical software R on a 2014 laptop with an Intel quad-core processor (2.70GHz, 2701Mhz, 8 logical processors, 8GB RAM). DTW was computed using the DTW R package [26]. Training the DTW-NN (calculating variable weights, DTW distance between reference batches throughout their durations, and control limits) took roughly 30 seconds. The elapsed real time for computing the online DTW distances between one new batch, and the 50 reference batches throughout the duration of the new batch was 0.7 seconds. To do this for all 2620 batches in one of the four datasets, parallel processing using three cores resulted in an elapsed real time of 10.1 minutes.

4. Conclusion

We have presented a novel data-driven online batch process monitoring tool, DTW-NN. This method was applied to four extensive datasets from a simulated penicillin production, and shown to have several advantages over the benchmark MPCA approach. DTW-NN can effectively deal with clustering of batches, and is intuitively simple in providing a direct similarity measure between an ongoing batch and previously observed NOC batches. In addition, the identification of the k nearest NOC batches provides the basis for an easy to interpret visualisation tool for process operators, enabling online comparison between the present batch and its most similar predecessors. A representative database of NOC batches is required for DTW-NN. As new NOC batches are completed, they may be added to the reference database which would lead to improvements in monitoring performance. In order to scale up DTW-NN to be efficient with searching a reference database of thousands or hundreds of thousands of batches, further research may be needed, for example making use of lower bounding and other speed-up techniques [27, 28]. However, DTW-NN as presented in this work may be readily implemented for effectively monitoring an ongoing batch in relation to a database of hundreds of past batches.

Acknowledgements

This research is partially funded by BIOPRO (www.biopro.nu) which is financed by the European Regional Development Fund (ERDF), Region Zealand (Denmark) and BIOPRO partners. The authors would like to thank Anders Stockmarr for constructive criticism of the manuscript.

References

[1] P. Nomikos, J. F. MacGregor, Monitoring Batch Processes Using Multiway Principal Component Analysis, AIChE Journal 40 (1994) 1361–1375.

[2] S. Wold, N. Kettaneh, H. Fridén, A. Holmberg, Modelling and diagnostics of batch processes and analogous kinetic experiments, Chemometrics and Intelligent Laboratory Systems 44 (1998) 331–340.

[3] J. A. Westerhuis, T. Kourti, J. F. MacGregor, Comparing alternative approaches for multivariate statistical analysis of batch process data, Journal of Chemometrics 13 (1999) 397–413.

[4] D. Neogi, C. E. Schlags, Multivariate statistical analysis of an emulsion batch process, Industrial and Engineering Chemistry Research 37 (1998) 3971–3979.

[5] S. García-Muñoz, T. Kourti, J. F. MacGregor, Troubleshooting of an Industrial Batch Process Using Multivariate Methods, Industrial & Engineering Chemistry Research (2003) 3592–3601.

[6] N. P. V. Nielsen, J. M. Carstensen, J. Smedsgaard, Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping, Journal of Chromatography A 805 (1998) 17–35.

[7] M. Fransson, S. Folestad, Real-time alignment of batch process data using COW for on-line process monitoring, Chemometrics and Intelligent Laboratory Systems 84 (2006) 56–61.

[8] A. Kassidas, J. F. MacGregor, P. A. Taylor, Synchronization of batch trajectories using dynamic time warping, AIChE Journal 44 (1998) 864–875.

[9] J. M. González-Martínez, A. Ferrer, J. A. Westerhuis, Real-time synchronization of batch trajectories for on-line multivariate statistical process control using Dynamic Time Warping, Chemometrics and Intelligent Laboratory Systems 105 (2011) 195–206.

[10] J. M. González-Martínez, O. E. De Noord, A. Ferrer, Multisynchro: A novel approach for batch synchronization in scenarios of multiple asynchronisms, Journal of Chemometrics 28 (2014) 462–475.

[11] R. Rendall, B. Lu, I. Castillo, S.-T. Chin, L. H. Chiang, M. S. Reis, A unifying and integrated framework for feature oriented analysis of batch processes, Industrial & Engineering Chemistry Research 56 (2017) 8590–8605.

[12] H. Sakoe, S. Chiba, Dynamic Programming Algorithm Optimization for Spoken Word Recognition, IEEE Transactions on Acoustics, Speech, and Signal Processing 26 (1978) 43–49.

[13] A. Mueen, E. Keogh, Extracting Optimal Performance from Dynamic Time Warping, in: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16, ACM Press, 2016, pp. 2129–2130. URL: http://dl.acm.org/citation.cfm?doid=2939672.2945383. doi:10.1145/2939672.2945383.

[14] Q. P. He, J. Wang, Fault detection using the k-nearest neighbor rule for semiconductor manufacturing processes, IEEE Transactions on Semiconductor Manufacturing 20 (2007) 345–354.

[15] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, E. Keogh, Experimental comparison of representation methods and distance measures

for time series data, Data Mining and Knowledge Discovery 26 (2013) 275–309.

[16] J. Van Impe, G. Gins, An extensive reference dataset for fault detection and identification in batch processes, Chemometrics and Intelligent Laboratory Systems 148 (2015) 20–31.

[17] T. Hastie, R. Tibshirani, J. Friedman, The Elements of Statistical Learning, volume 1, Springer New York, 2001. URL: http://www.springerlink.com/index/10.1007/b94608. doi:10.1007/b94608. arXiv:arXiv:1011.1669v3.

[18] H. J. Ramaker, E. N. M. Van Sprang, J. A. Westerhuis, A. K. Smilde, Dynamic time warping of spectroscopic BATCH data, Analytica Chimica Acta 498 (2003) 133–153.

[19] S. García-Muñoz, M. Polizzi, A. Prpich, C. Strain, A. Lalonde, V. Negron, Experiences in batch trajectory alignment for pharmaceutical process improvement through multivariate latent variable modelling, Journal of Process Control 21 (2011) 1370–1377.

[20] M. Spooner, D. Kold, M. Kulahci, Selecting local constraint for alignment of batch process data with dynamic time warping, Chemometrics and Intelligent Laboratory Systems 167 (2017) 161–170.

[21] L. Kaufman, P. J. Rousseeuw, Finding groups in data: an introduction to cluster analysis, Wiley, New York, 1990. doi:doi:10.1002/9780470316801.fmatter.

[22] P. Nomikos, J. F. MacGregor, Multivariate SPC charts for monitoring batch processes, Technometrics 37 (1995) 41–59.

[23] T. J. Rato, R. Rendall, V. Gomes, S. T. Chin, L. H. Chiang, P. M. Saraiva, M. S. Reis, A Systematic Methodology for Comparing Batch Process Monitoring Methods: Part I-Assessing Detection Strength, Industrial and Engineering Chemistry Research 55 (2016) 5342–5358.

[24] D. J. Louwerse, A. K. Smilde, Multivariate statistical process control of batch processes based on three-way models, Chemical Engineering Science 55 (2000) 1225–1235.

[25] G. Birol, C. Ündey, A. Çinar, A modular simulation package for fed-batch fermentation: Penicillin production, Computers and Chemical Engineering 26 (2002) 1553–1565.

[26] T. Giorgino, Computing and Visualizing Dynamic Time Warping Alignments in R : The dtw Package, Journal of Statistical Software 31 (2009) 1–24.

[27] E. Keogh, C. A. Ratanamahatana, Exact indexing of dynamic time warping, Knowledge and Information Systems 7 (2005) 358–386.

[28] T. Rakthanmanon, B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, E. Keogh, Searching and mining trillions of time series subsequences under dynamic time warping, Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12 (2012) 262.

CHAPTER 7

Conclusion

The motivation for this PhD research was to investigate methods for exploiting the ample production data produced in bio-manufacturing processes. Collaboration with two industry partners, Chr. Hansen and CP Kelco, confirmed that there were indeed opportunities for improved utilisation of production data in real industrial settings, leading to the contributions that have been presented in this thesis.

A major theme in the key findings of this thesis has been the alignment of process data using DTW. The existing literature on statistical analysis of batch processes often used DTW as a pre-processing step to obtain variable trajectories of equal length prior to analysis with, for example, PCA. The data encountered in the project with Chr. Hansen displayed considerable time variation, with the same biological event occurring at different times in different batches. It was observed that, due to their dependence on living organisms, time variation plays a big role in bio-processes. A need was identified for adapting DTW so that the resulting warping function would be a more realistic representation of the progress signature of a batch. Existing research had focused on global constraints for limiting DTW in batch process alignment. It was demonstrated that local constraints provide a more appropriate way of adjusting the amount of warping, and a novel method for selecting the local constraint has been presented.

The second project with Chr. Hansen showed how the warping function from DTW could be used to make online predictions of the final harvest time of batches. Lasso regression was used to predict harvest time based on the first phase of the process, then during the second phase, online DTW was used to update the predictions, and the RMSE was observed to decrease over time. This was possible because the online DTW warping function provides an indication of how fast or slow the batch is progressing.

The final DTW based finding was the new monitoring method presented in Chapter 6. This method uses the DTW distance to find Nearest Neighbour batches to an ongoing batch, and thereby determine if the ongoing batch is in control or not. The method was shown to have a greater true detection rate than conventional MPCA monitoring when applied to a simulated dataset containing clustering between batches. Another advantage of the DTW Nearest Neighbour approach is that it identifies the most similar previous batch to the ongoing batch. This is useful information for process operators. In bio-manufacturing, it was observed that human expertise played a direct role in overseeing processes, which in some ways often resembled as much an art as a science. Highlighting that an ongoing batch resembles closely, for example, a specific batch completed last Tuesday may have significant meaning to process operators. Used in this way, the nearest neighbour approach could supplement the human expertise in the process by providing directly relateable information, rather than simply signalling or not signalling an alarm.

Interpretability of models has been a secondary theme throughout the thesis. Whether predicting harvest time (Chapter 4) or quality parameters (Chapter 5) in real batch processes, it was found that lasso regression provided far more parsimonious and interpretable models than the more commonly applied PLS regression, whilst performing just as well in terms of estimated prediction error. In the unfolded batch process data, it is unreasonable to expect every variable at every time-point to be relevant for the response variable of interest, and so the sparse lasso method was found to be preferable to the latent structure PLS method. In the chemometrics setting, PLS is typically used to relate spectroscopic data to some quality measurement, and prior chemical knowledge is often used to narrow the input data to a range of wavelengths which is suspected to be of relevance. In this setting, PLS is a natural choice. However, the problem confronted in predicting quality from batch process engineering variables is quite different. The input variables typically vary greatly in nature, and there is less prior knowledge of the relationships that may be present. In this, more exploratory setting, a method which includes variable selection is more attractive. The case studies in this thesis have shown that lasso regression is a useful alternative to PLS.

Other findings of importance relate to the challenges encountered in the industrial case studies. These contrasted both with assumptions often made in the academic literature, as well as with some of the expectations generated by the publicity of big data. A big assumption made in most of the literature on batch processes, is the availability of a large database of comparable batches from normal operating conditions on which to base a monitoring scheme. However, as batch processes are a mode of production especially suited to small runs of speciality products, and frequent changing of products, often sufficient data is difficult to obtain. Indeed, this challenge was encountered during collaboration with Chr. Hansen, as discussed in Chapter 3. From these experiences, it is concluded that greater focus is needed on developing, for example, visualisation tools that can facilitate the existing human

expertise present in the process, rather than new monitoring models that depend on numerous comparable batches for validity. A step in this direction has been provided with the DTW nearest neighbour scheme presented in Chapter 6. Granted, it is formulated in the context of traditional monitoring, but as noted in the paper, it contains the framework for providing the visualisation of an ongoing batch with the most similar previously encountered batches, even if only a few previous batches are available.

The publicity of big data has raised expectations in many fields of data analysis, and indeed influenced the motivation behind this thesis. From the presented work, a greater understanding has been obtained on what can realistically be achieved using industrial production data. A myth that seems to accompany the big data narrative, is that with a sufficiently large amount of data, and a sufficiently advanced algorithm, there is no limit to what can be accurately predicted. Of course, in practice success depends also on the type of data available. In the case of predicting quality parameters from production data, the challenge is that real production is typically designed to operate within narrow bands of process conditions and product quality. This limits what causal relationships can be discovered, no matter how much production data is available. This challenge was encountered in the collaboration with CP Kelco presented in Chapter 5. It was not possible to obtain a model that could predict DA and DE with sufficient accuracy to replace the need for measuring the quality parameters. However, some new insight was obtained from the project, including which process conditions may have greater importance for the quality parameters. There is certainly value to be found in the analysis of production data, but to ensure a satisfactory outcome it is important for research and industry to understand what may be realistically achieved. In this work it has been found that production data is suited to exploratory investigation of the process rather than development of directly implementable prediction models.

There are a number of points in which the work of this thesis may be extended. One aspect of DTW related to batch processes which seems especially open is the weighting of variables for calculation of the local distance ($\mathbf{W}$ in Eq. 2.17). If the goal is to obtain the most accurate alignment so that the warping function is a good representation of the progress of a batch, than the method proposed by Ramaker et al. (2003) seems adequate as discussed in Paper 1. However, in Paper 3, it was the aim that the DTW distance itself should be sensitive to abnormalities in all of the variables, and therefore a different weighting method, effectively scaling by each variable's measurement noise, was adopted. Further research is needed to resolve the weighting of variables in DTW, and to optimise multidimensional DTW in general, especially with regards to the curse of dimensionality.

Another direction for further research would be towards making the nearest neighbour search used in Paper 3 more efficient. In that work, an exhaustive search is performed at each new time point, requiring the calculation of the DTW distance

between the ongoing batch so-far, and every complete batch in the database. A possibility for reducing the computational load, and perhaps noise in the monitoring statistic, would be to only consider a change in the members of the k nearest neighbours if the distance to the current nearest neighbours increases by "too much". Alternatively, the lower bounding method presented by Keogh and Ratanamahatana (2005) and Rakthanmanon et al. (2012) could be pursued, which provides a simple lower bound to the magnitude of the DTW distance. This allows many time series in a database to be quickly excluded as possible nearest neighbours without having to calculate the exact DTW distance, speeding up the search. A limitation of the method which must be overcome for online distance calculation of batches is that the lower bound only holds for comparing time series of the same length.

The topic of fault diagnosis was not addressed in Paper 3, which focused only on fault detection, but is a very relevant issue. Determining what has gone wrong, that is, which variables are principally to blame for the out of control signal, is necessary for taking corrective action after a fault has been detected. Research into diagnosis or classification of detected faults based on DTW is a natural extension of Paper 3.

The potential of lasso regression for prediction from batch process data is another area which merits further investigation. For example, are there advantages to using elastic net regression, rather than lasso regression, for batch processes? Another extension of the lasso which would seem to lend itself to the unfolded batch process data is the group lasso developed by Yuan and Lin (2006). This method allows predefined groups of variables to be selected or excluded from the model together. This could be applied to the unfolded data so that if a variable is included, it is included at all time-points, or else left out entirely, and may result in a more natural model. Finally, it would be of great interest to tackle the strict linearity of lasso regression models by considering interactions between variables. This presents the computational challenge that for p variables, there are $\binom{p}{2} = \frac{1}{2}p(p-1)$ possible two way interactions to consider, so the model matrix formed by adding two-way interactions to the unfolded batch process data may be intractable. Another issue to consider is whether to preserve model hierarchy in the way the lasso model selects or excludes interactions, and a way to do this is presented by Lim and Hastie (2015).

In addition to these interesting possibilities for future research topics, it must be emphasized in closing, that for progress to be made in the utilisation of data in the bio-manufacturing industry, sharing of experiences using real case studies is essential. To this end it is hoped that the work presented in this thesis provides new insights to both academic and industrial practitioners in the bio-manufacturing community.

Bibliography

Alfke, G., W. W. Irion, and O. S. Neuwirthe. 2007. *Oil Refining*. doi:10.1002/14356007.a18.

Birol, G., C. Ündey, and A. Çinar. 2002. "A modular simulation package for fed-batch fermentation: Penicillin production". *Computers and Chemical Engineering* 26 (11): 1553–1565. ISSN: 00981354. doi:10.1016/S0098-1354(02)00127-8.

Boqué, R., and A. K. Smilde. 1999. "Monitoring and diagnosing batch processes with multiway covariates regression models". *AIChE Journal* 45 (7): 1504–1520. ISSN: 00011541. doi:10.1002/aic.690450713.

Chen, J., and K. Liu. 2002. "On-line batch process monitoring using dynamic PCA and dynamic PLS models". *Chemical Engineering Science* 57:63–75.

Chiu, C.-c., and Y. Yao. 2013. "Multiway elastic net (MEN) for final product quality prediction and quality-related analysis of batch processes". *Chemometrics and Intelligent Laboratory Systems* 125:153–165. ISSN: 01697439. doi:10.1016/j.chemolab.2013.04.006. http://dx.doi.org/10.1016/j.chemolab.2013.04.006.

Chong, I. G., and C. H. Jun. 2005. "Performance of some variable selection methods when multicollinearity is present". *Chemometrics and Intelligent Laboratory Systems* 78 (1): 103–112. ISSN: 01697439. doi:10.1016/j.chemolab.2004.12.011.

Chr. Hansen A/S. 2018. *Chr. Hansen: About Us*. https://www.chr-hansen.com/en/about-us.

Croughan, M. S., K. B. Konstantinov, and C. Cooney. 2015. "The future of industrial bioprocessing: Batch or continuous?" *Biotechnology and Bioengineering* 112 (4): 648–651. ISSN: 10970290. doi:10.1002/bit.25529.

Desai, K., Y. Badhe, S. S. Tambe, and B. D. Kulkarni. 2006. "Soft-sensor development for fed-batch bioreactors using support vector regression". *Biochemical Engineering Journal* 27 (3): 225–239. ISSN: 1369703X. doi:10.1016/j.bej.2005.08.002.

Desai, K., B. K. Vaidya, R. S. Singhal, and S. S. Bhagwat. 2005. "Use of an artificial neural network in modeling yeast biomass and yield of β-glucan". *Process Biochemistry* 40 (5): 1617–1626. ISSN: 13595113. doi:10.1016/j.procbio.2004.06.015.

Felder, R., and R. Rousseau. 2003. *Elementary Principles of Chemical Processes*. 3rd edition. New York: Wiley. ISBN: 978-0-471-37587-6.

Flutto, L., and Danisco. 2003. "Pectin". *Encyclopedia of Food Sciences and Nutrition*, number 1998: 4440–4449. doi:10.1016@B0-12-227055-X@00901-9.

Frank, I. E., and J. Friedman. 1993. "A Statistical View of Some Chemometrics View Regression Tools". *Technometrics* 35 (2): 109–135. ISSN: 0040-1706. doi:10.2307/1269656.

Fransson, M., and S. Folestad. 2006. "Real-time alignment of batch process data using COW for on-line process monitoring". *Chemometrics and Intelligent Laboratory Systems* 84 (1-2 SPEC. ISS.): 56–61. ISSN: 01697439. doi:10.1016/j.chemolab.2006.04.020.

García-Muñoz, S., T. Kourti, and J. F. MacGregor. 2004. "Model predictive monitoring for batch processes". *Industrial and Engineering Chemistry Research* 43 (18): 5929–5941. ISSN: 08885885. doi:10.1021/ie034020w.

— . 2003. "Troubleshooting of an Industrial Batch Process Using Multivariate Methods". *Industrial & Engineering Chemistry Research*, number 42: 3592–3601.

García-Muñoz, S., M. Polizzi, A. Prpich, C. Strain, A. Lalonde, and V. Negron. 2011. "Experiences in batch trajectory alignment for pharmaceutical process improvement through multivariate latent variable modelling". *Journal of Process Control* 21 (10): 1370–1377. ISSN: 09591524. doi:10.1016/j.jprocont.2011.07.013.

Geladi, P., and B. R. Kowalski. 1986. "Partial least-squares regression: a tutorial". *Analytica Chimica Acta* 185 (C): 1–17. ISSN: 00032670. doi:10.1016/0003-2670(86)80028-9. arXiv: arXiv:1011.1669v3.

González-Martínez, J. M., A. Ferrer, and J. A. Westerhuis. 2011. "Real-time synchronization of batch trajectories for on-line multivariate statistical process control using Dynamic Time Warping". *Chemometrics and Intelligent Laboratory Systems* 105 (2): 195–206. ISSN: 01697439. doi:10.1016/j.chemolab.2011.01.003. http://dx.doi.org/10.1016/j.chemolab.2011.01.003.

González-Martínez, J. M., R. Vitale, O. E. De Noord, and A. Ferrer. 2014. "Effect of synchronization on bilinear batch process modeling". *Industrial and Engineering Chemistry Research* 53 (11): 4339–4351. ISSN: 15205045. doi:10.1021/ie402052v.

Gunther, J. C., J. S. Conner, and D. E. Seborg. 2009. "Process monitoring and quality variable prediction utilizing PLS in industrial fed-batch cell culture". *Journal of Process Control* 19 (5): 914–921. ISSN: 09591524. doi:10.1016/j.jprocont.2008.11.007. http://dx.doi.org/10.1016/j.jprocont.2008.11.007.

Hastie, T., R. Tibshirani, and J. Friedman. 2001. *The Elements of Statistical Learning*. 1:1–694. Springer New York. ISBN: 978-0-387-84857-0. doi:10.1007/b94608. arXiv: arXiv:1011.1669v3. http://www.springerlink.com/index/10.1007/b94608.

Hoerl, A. E., and R. W. Kennard. 1970. "Ridge Regression: Biased Estimation for Nonorthogonal Problems". *Technometrics* 12 (1): 55–67. ISSN: 0040-1706. doi:10.1080/00401706.2000.10485983.

Kassidas, A., J. F. MacGregor, and P. A. Taylor. 1998. "Synchronization of batch trajectories using dynamic time warping". *AIChE Journal* 44 (4): 864–875. ISSN: 00011541. doi:10.1002/aic.690440412. http://doi.wiley.com/10.1002/aic.690440412.

Kelco, C. P. 2015. *Portræt CP Kelco, Lille Skensved*. Technical report. http://www.cpkelco.com/about-cp-kelco/our-company/lille-skensved/.

Keogh, E., and C. A. Ratanamahatana. 2005. "Exact indexing of dynamic time warping". *Knowledge and Information Systems* 7, number 3 (): 358–386. ISSN: 0219-3116. doi:10.1007/s10115-004-0154-9. https://doi.org/10.1007/s10115-004-0154-9.

Klimkiewicz, A. 2016. "Multivariate Statistical Process Optimization in the Industrial Production of Enzymes". PhD thesis, University of Copenhagen.

Kourti, T. 2003. "Abnormal situation detection, three-way data and projection methods; robust data archiving and modeling for industrial applications". *Annual Reviews in Control* 27 II:131–139. ISSN: 13675788. doi:10.1016/j.arcontrol.2003.10.004.

Ku, W., R. H. Storer, and C. Georgakis. 1995. "Disturbance detection and isolation by dynamic principal component analysis". *Chemometrics and Intelligent Laboratory Systems* 30 (1): 179–196. ISSN: 01697439. doi:10.1016/0169-7439(95)00076-3.

Lee, J. M., C. K. Yoo, and I. B. Lee. 2004. "Enhanced process monitoring of fed-batch penicillin cultivation using time-varying and multivariate statistical analysis". *Journal of Biotechnology* 110 (2): 119–136. ISSN: 01681656. doi:10.1016/j.jbiotec.2004.01.016.

Lim, M., and T. Hastie. 2015. "Learning Interactions via Hierarchical Group-Lasso Regularization". *Journal of Computational and Graphical Statistics* 24 (3): 627–654. ISSN: 15372715. doi:10.1080/10618600.2014.938812. arXiv: 1308.2719.

Louwerse, D. J., and A. K. Smilde. 2000. "Multivariate statistical process control of batch processes based on three-way models". *Chemical Engineering Science* 55 (7): 1225–1235. ISSN: 00092509. doi:10.1016/S0009-2509(99)00408-X. http://www.sciencedirect.com/science/article/pii/S000925099900408X.

Lu, B., S. Xu, J. Stuber, and T. F. Edgar. 2016. "Constrained selective dynamic time warping of trajectories in three dimensional batch data". *Chemometrics and Intelligent Laboratory Systems* 159 (October): 138–150. ISSN: 18733239. doi:10.1016/j.chemolab.2016.10.005.

Lu, N., and F. Gao. 2005. "Stage-Based Process Analysis and Quality Prediction for Batch Processes". *Industrial & Engineering Chemistry Research* 44 (10): 3547–3555. ISSN: 0888-5885, 1520-5045. doi:10.1021/ie048852l. http://pubs.acs.org/doi/abs/10.1021/ie048852l.

Luo, L., S. Bao, Z. Gao, and J. Yuan. 2014. "Batch process monitoring with GTucker2 model". *Industrial and Engineering Chemistry Research* 53 (39): 15101–15110. ISSN: 15205045. doi:10.1021/ie5015102.

Montgomery, D. C. 2013. *Statistical Quality Control*. 7th edition. Wiley. ISBN: 978-1-118-32257-4.

Nielsen, N. P. V., J. M. Carstensen, and J. Smedsgaard. 1998. "Aligning of single and multiple wavelength chromatographic profiles for chemometric data analysis using correlation optimised warping". *Journal of Chromatography A* 805 (1-2): 17–35. ISSN: 00219673. doi:10.1016/S0021-9673(98)00021-1.

Nomikos, P., and J. F. MacGregor. 1994. "Monitoring Batch Processes Using Multiway Principal Component Analysis". *AIChE Journal* 40 (8): 1361–1375. ISSN: 0001-1541. doi:10.1002/aic.690400809.

— . 1995a. "Multi-way partial least squares in monitoring batch processes". *Chemometrics and Intelligent Laboratory Systems* 30 (1): 97–108. ISSN: 01697439. doi:10.1016/0169-7439(95)00043-7.

— . 1995b. *Multivariate SPC charts for monitoring batch processes*. doi:10.1080/00401706.1995.10485888.

Onel, M., C. A. Kieslich, Y. A. Guzman, C. A. Floudas, and E. N. Pistikopoulos. 2018. "Big data approach to batch process monitoring: Simultaneous fault detection and diagnosis using nonlinear support vector machine-based feature selection". *Computers and Chemical Engineering* 115:46–63. ISSN: 00981354. doi:10.1016/j.compchemeng.2018.03.025. https://doi.org/10.1016/j.compchemeng.2018.03.025.

Rakthanmanon, T., B. Campana, A. Mueen, G. Batista, B. Westover, Q. Zhu, J. Zakaria, and E. Keogh. 2012. "Searching and mining trillions of time series subsequences under dynamic time warping". *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '12*: 262. ISSN: 10450823. doi:10.1145/2339530.2339576. http://dl.acm.org/citation.cfm?doid=2339530.2339576.

Ramaker, H.-J., E. N. Van Sprang, J. A. Westerhuis, and A. K. Smilde. 2003. "Dynamic time warping of spectroscopic BATCH data". *Analytica Chimica Acta* 498 (1-2): 133–153. ISSN: 00032670. doi:10.1016/j.aca.2003.08.045.

— . 2005. "Fault detection properties of global, local and time evolving models for batch process monitoring". *Journal of Process Control* 15 (7): 799–805. ISSN: 09591524. doi:10.1016/j.jprocont.2005.02.001.

Rato, T. J., R. Rendall, V. Gomes, P. M. Saraiva, and M. S. Reis. 2018. "A Systematic Methodology for Comparing Batch Process Monitoring Methods: Part II—Assessing Detection Speed". *Industrial & Engineering Chemistry Research*: acs.iecr.7b04911. ISSN: 0888-5885. doi:10.1021/acs.iecr.7b04911. http://pubs.acs.org/doi/10.1021/acs.iecr.7b04911.

Rato, T., R. Rendall, V. Gomes, S. T. Chin, L. H. Chiang, P. M. Saraiva, and M. S. Reis. 2016. "A Systematic Methodology for Comparing Batch Process Monitoring Methods: Part I-Assessing Detection Strength". *Industrial and Engineering Chemistry Research* 55 (18): 5342–5358. ISSN: 15205045. doi:10.1021/acs.iecr.5b04851.

Sakoe, H., and S. Chiba. 1978. "Dynamic Programming Algorithm Optimization for Spoken Word Recognition". *IEEE Transactions on Acoustics, Speech, and Signal Processing* 26 (1): 43–49. ISSN: 00963518. doi:10.1109/TASSP.1978.1163055. arXiv: arXiv:1011.1669v3.

Shewhart, W. A. 1931. *Economic control of quality of manufactured product*. ASQ Quality Press.

Shokoohi-Yekta, M., J. Wang, and E. Keogh. 2015. "On the Non-Trivial Generalization of Dynamic Time Warping to the Multi-Dimensional Case". *Proceedings of the 2015 SIAM International Conference on Data Mining*: 289–297. ISSN: 978-1-61197-401-0. doi:10.1137/1.9781611974010.33. http://epubs.siam.org/doi/10.1137/1.9781611974010.33.

Skov, T., F. Van den Berg, G. Tomasi, and R. Bro. 2007. "Automated alignment of chromatographic data". *Journal of Chemometrics* 20:484–497. ISSN: 1099-128X. doi:10.1002/cem.1031.

Spooner, M., D. Kold, and M. Kulahci. 2018. "Harvest time prediction for batch processes". *Computers & Chemical Engineering* 117:32–41. ISSN: 0098-1354. doi:https://doi.org/10.1016/j.compchemeng.2018.05.019. https://www.sciencedirect.com/science/article/pii/S0098135418305039.

— . 2017. "Selecting local constraint for alignment of batch process data with dynamic time warping". *Chemometrics and Intelligent Laboratory Systems* 167:161–170. ISSN: 18733239. doi:10.1016/j.chemolab.2017.05.019. http://dx.doi.org/10.1016/j.chemolab.2017.05.019.

Spooner, M., and M. Kulahci. 2018. "Monitoring batch processes with dynamic time warping and k-nearest neighbours". *Chemometrics and Intelligent Laboratory Systems*. doi:https://doi.org/10.1016/j.chemolab.2018.10.011.

Thornhill, N. F., M. A. Shoukat Choudhury, and S. L. Shah. 2004. "The impact of compression on data-driven process analyses". *Journal of Process Control* 14 (4): 389–398. ISSN: 09591524. doi:10.1016/j.jprocont.2003.06.003.

Tibshirani, R. 1996. "Regression Shrinkage and Selection via the Lasso". *Journal of the Royal Statistical Society. Series B (Methodological)* 58 (1): 267–288. http://www.jstor.org/stable/2346178.

Ündey, C., and A. Çinar. 2002. "Statistical Monitoring of Multistage, Multiphase Batch Processes". *IEEE Control Systems* 22 (5): 40–52. ISSN: 1066033X. doi:10.1109/MCS.2002.1035216.

Ündey, C., S. Ertunç, and A. Çinar. 2003. "Online Batch/Fed-Batch Process Performance Monitoring, Quality Prediction, and Variable-Contribution Analysis for Diagnosis". *Industrial & Engineering Chemistry Research* 42 (20): 4645–4658. ISSN: 0888-5885. doi:10.1021/ie0208218. http://dx.doi.org/10.1021/ie0208218%7B%5C%%7D5Cnhttp://pubs.acs.org/doi/full/10.1021/ie0208218%7B%5C%%7D5Cnhttp://pubs.acs.org/doi/pdf/10.1021/ie0208218.

Van Impe, J. F., and G. Gins. 2015. "An extensive reference dataset for fault detection and identification in batch processes". *Chemometrics and Intelligent Laboratory Systems* 148:20–31. ISSN: 18733239. doi:10.1016/j.chemolab.2015.08.019. http://dx.doi.org/10.1016/j.chemolab.2015.08.019.

Westerhuis, J. A., T. Kourti, and J. F. MacGregor. 1999. "Comparing alternative approaches for multivariate statistical analysis of batch process data". *Journal of Chemometrics* 13 (3-4): 397–413. ISSN: 1099-128X. doi:10.1002/(SICI)1099-128X(199905/08)13:3/4<397::AID-CEM559>3.0.CO;2-I. http://www.scopus.com/inward/record.url?eid=2-s2.0-0000609488%7B%5C&%7DpartnerID=40%7B%5C&%7Dmd5=1dc8c89b8ac8c4de3a59b64400b2d237.

Wise, B. M., N. B. Gallagher, and E. B. Martin. 2001. "Application of PARAFAC2 to fault detection and diagnosis in semiconductor etch". *Journal of Chemometrics* 15 (4): 285–298. ISSN: 08869383. doi:10.1002/cem.689.

Wold, S., N. Kettaneh, H. Fridén, and A. Holmberg. 1998. "Modelling and diagnostics of batch processes and analogous kinetic experiments". *Chemometrics and Intelligent Laboratory Systems* 44 (1-2): 331–340. ISSN: 01697439. doi:10.1016/S0169-7439(98)00162-2.

Yan, Z., C.-c. Chiu, Y. Yao, and W. Doing. 2014. "Regularization-based statistical batch process modeling for final product quality prediction". *AIChE Journal* 60 (504): 2815–2827. ISSN: 12350621. doi:10.1002/aic. arXiv: 0201037v1 [arXiv:physics].

Yuan, M., and Y. Lin. 2006. "Model Selection and Estimation in Regression with Grouped Variables". *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 68 (1): 49–67.

Zou, H., and T. Hastie. 2005. "Regularization and variable selection via the elastic net". *Journal of the Royal Statistical Society. Series B: Statistical Methodology* 67 (2): 301–320. ISSN: 13697412. doi:10.1111/j.1467-9868.2005.00503.x.